NATIONAL BUREAU OF STANDARDS REPORT

10 436

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix B

PROGRAM LISTINGS FOR PROGRAMMER LEVEL DOCUMENTATION

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics [2]—Radio Engineering [2]—Time and Frequency [2]—Astrophysics [2]—Cryogenics.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures — Invention and Innovation — Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information — Computer Services — Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information [3]—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT
4314561

NBS REPORT
10 436

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix B

PROGRAM LISTINGS FOR PROGRAMMER LEVEL DOCUMENTATION

IMPORTANT NOTICE

NATIONAL BUREAU OF ST ... ss accounting documents intended for use within the Government. ... subjected to additional evaluation and review. For this reason, th ... listing of this Report, either in whole or in part, is not autho ... e Office of the Director, National Bureau of Standards, Washingt ... by the Government agency for which the Report has been specificall ... copies for its own use.

Approved for public release by the Director of the National Institute of Standards and Technology (NIST) on October 9, 2015.

NBS

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

PREFACE

This volume is one of a series which documents a Search and Rescue Simulation Model for the United States Coast Guard. The material reported in this documentation was developed by an interdisciplinary team at the National Bureau of Standards with representation from the U.S. Coast Guard under MIPR Z-70099-0-01935.

The complete documentation is comprised of the following:

Volume I	Executive Level Documentation
Volume II	Analyst Level Documentation
Volume III	Programmer Level Documentation for "PREPROCESSOR"
Volume IV	Programmer Level Documentation for "OPSIM"
Volume V	Programmer Level Documentation for "POSTPROCESSOR"
Appendix A	Flow Charts for Programmer Level Documentation
Appendix B	Program Listings for Programmer Level Documentation

The study was initially conducted under the supervision of Martin J. Aronoff; subsequently efforts were supervised by Richard T. Penn, Jr. Technical Project Leadership was supplied throughout the project by Stephen S. Karp. Other participants from the National Bureau of Standards Technical Analysis Division included the following:

Susan S. Chamberlin	Elizabeth E. Leyendecker
Linda K. Cummings	Marcia D. Maltese
Mary Jane Duberg	Patsy L.B. Saunders*
William Elliott, III	Wayne A. Steele
Walter G. Leight	Michael R. Vogt
Joel Levy	Arnold L. Weber

Valuable advice was received from Alan J. Goldman* and Prof. Gustave J. Rath of Northwestern University.

U.S. Coast Guard participants included:

Paul D'Zmura	Gerald L. Underwood
Thomas T. Matteson	Robert R. Wells

Support services were furnished by the following members of the NBS Technical Analysis Divison:

Mary M. Abbott	Frances E. Jones
Theresa I. Conrad	Lucinda I. Farrell

* Staff members of the NBS Applied Mathematics Divison

Appendix B

Table of Contents

page

PREPRO

MUTAPE 1

MAIN PROGRAM 1A

READ 7

WRITE 9

MUC130 11

MAIN PROGRAM 12

READ 16

WRITE 18

PCP 20

MAIN PROGRAM 20

READ 30

NEEDS 32

NUCASE 34

FIELD 37

RAND 41

DEMGEN 42

IO TABLE 42

MAIN PROGRAM 44

BOX 55

ADD 57

ADTIME 59

RAND 61

SELECT 62

HIST 64

MAIN PROGRAM 65

CONVER 69

OPSIM 71

DEFINITIONS 71

EVENTS 75

START 76

NVCRU 77

OPSIM 80

SRAS 84

CRES 85

VEC 89

OSET 90

RESAP 91

ROCA 94

MRAS 95

NOTIF 96

Table of Contents (con't.)

	page
TOW	97
DTD	98
SRCH	99
NOTE	101
SASS	102
SSS	104
READY	108
ARSCH	109
COMPL	111
SSET	112
FUEL	113
HOMEF	114
SNDBK	115
SRISE	117
XSET	118
SERVE	119
DELAY	121
ARVSN	122
STATS	125
SRCHF	127
ONSCN	129
COVER	132
CHEKN	133
RETN	134
TERM	135
HOME	137
SAQ	138
QUEUE	140
EXQ	142
STNBY	147
SVQUE	151
WRECK	152
ENDSIM	154
DRIVE	157
REPORT SARSIM	159
REPORT GRPRES	162
REPORT HEADER	163
REPORT TITLE	164
REPORT RESULT	165
REPORT HEAD	166
EXCASE	167
REPORT DISTRIB	168
JUMPER	170

MUTAPE

This program is not completely parameterized. For each district run, one source card must be changed. In the main program, card number 8 includes the number for the district being exercised in columns 38 and 39. In the listing given the district is 3. For each run made this number must be set to equal the number of the district for which data are being processed.

```
@IT FOR MUTAPE,MUTAPE                                                                    03 JUN 71          22:51:11.692
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F5018P
THIS COMPILATION WAS DONE ON 03 JUN 71 AT 22:51:11


   MAIN PROGRAM

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    001764
         0000    *DATA    043317
         0002    *BLANK   003427
         0003    BREAD    000011
         0004    BWRITE   000011


   EXTERNAL REFERENCES (BLOCK, NAME)

         0005    CLOCKS
         0006    NTRAN
         0007    READ
         0010    WRITE
         0011    SOPEN2
         0012    SRREL
         0013    SSORT
         0014    SRRET
         0015    SOPEN3
         0016    NWDU$
         0017    NIO1$
         0020    NIO2$
         0021    NSTOP$
         0022    NREW$
         0023    NRBU$
         0024    NWBU$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000063 10L        0000    043226 100F       0000    043231 101F       0000    043173 102F       0001    000151 11L
    0001    000473 12L        0001    001433 13L        0001    001470 14L        0001    000055 142G       0001    001506 16L
    0001    000126 165G       0001    001650 17L        0000    043235 171F       0001    001667 172L       0001    000143 174G
    0001    001721 18L        0001    001756 19L        0001    000200 20L        0001    000213 221G       0001    000253 234G
    0001    000421 24L        0001    000267 241G       0001    000316 254G       0001    000463 26L        0001    000362 266G
    0001    000470 27L        0001    000477 30L        0001    000410 300G       0001    000501 301L       0001    000540 31L
    0001    000442 313G       0001    000453 321G       0001    000603 33L        0001    000622 34L        0000    043202 341F
    0001    000660 342L       0000    043214 343F       0001    000556 344G       0001    000701 344L       0001    000720 35L
    0001    000567 351G       0001    000742 37L        0001    001005 40L        0001    001126 411L       0001    000714 416G
    0001    001157 42L        0001    000725 425G       0001    001225 44L        0001    000766 441G       0001    001235 441L
    0001    000775 447G       0001    001032 463G       0001    001346 47L        0001    001050 473G       0001    001355 48L
    0001    001421 50L        0001    001105 506G       0001    001200 524G       0001    001211 531G       0001    001262 551G
    0001    001311 563G       0001    001335 575G       0001    001402 615G       0001    001413 623G       0001    001477 647G
    0001    001643 700G       0001    001710 725G       0000 I  043156 A3A        0000 I  043171 A4A        0000 I  043172 A4B
    0000 I  043166 B          0000 I  001762 BUFF       0000 I  043165 BX         0000 I  043170 BY         0000 I  043155 C
    0000 I  000012 CASE       0004 I  000004 CF         0000 I  043140 CODE       0000 I  043152 CX         0000 I  043161 D
    0000 I  000776 DATA       0004 I  000006 DP         0004 I  000005 DPN        0002 I  001000 DUMP       0000 I  043157 DX
    0002 R  002000 E          0000 I  043145 I          0000 I  043154 IEND       0003 I  000001 IF         0000 I  043144 IFLAG
```

1A

```
0002 I 002005 IHDR      0003 I 000003 INREX     0004 I 000003 IOREX     0002 I 003423 IPREC     0002 I 002003 IQT
0000 I 043167 IQUIT     0003 I 000000 IS        0000 I 043150 ISKIP     0003 I 000007 ISTAT     0002 I 002004 IUNIT
0000 I 043162 IX        0000 I 043141 I$        0000 I 043160 J         0002 I 002023 JHDR      0002 I 003424 JPREC
0003 I 000004 KF        0000 I 043164 KUCASF    0000 I 043147 KUDIST    0000 I 043146 KUOPFC    0000 I 043153 LAST
0000 I 043151 LGF       0000 I 043143 LGFS      0004 I 000002 LIM       0003 I 000002 LMT       0002 I 000000 LOAD
0000 I 003732 LONG      0003 I 000006 LP        0003 I 000005 LPN       0000 I 043142 LX        0002 I 003426 MPREC
0000 I 000000 MUHDR     0004 I 000001 NF        0000 I 013575 NLGF      0002 I 003425 NPREC     0004 I 000000 NS
0004 I 000007 NSTAT     0000 I 043163 NUCASF    0000 I 013600 NUMB      0000 I 013572 NX        0002 R 002001 XO
0002 R 002002 X1
```

```
00101      1*        IMPLICIT INTEGER (A,B,C,D)
00103      2*        COMMON LOAD(512),DUMP(512),E,XO,X1,IQT,IUNIT,IHDR(14),JHDR(768),
00103      3*       1      IPREC,JPREC,NPREC,MPREC
00104      4*        COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00105      5*        COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00106      6*        DIMENSION MUHDR(10),CASE(500),DATA(500),BUFF(1000),LONG(4000),
00106      7*       1      NX(3),NLGF(3),NUMB(3,4000)
00107      8*        DATA CODE /'@@@@@U'/, MUHDR /' 03                MULTI-UNIT       ',
00107      9*       1      '                          '/
00112     10*        E=600.0
00113     11*        CALL CLOCKS(XO)
00114     12*        XO=XO+E
00115     13*        CALL NTRAN (7,2,256,LOAD(1),ISTAT(1))
00116     14*        IS=2
00117     15*        IF=257
00120     16*        LMT=482
00121     17*        INREX=0
00122     18*        NPREC=0
00123     19*        MPREC=0
00124     20*        CALL READ (<50)
00125     21*        LP=IF+32
00126     22*        NS=1
00127     23*        NF=1
00130     24*        LIM=226
00131     25*        IOREX=0
00132     26*        DP=NF
00133     27*        NX(1)=0
00134     28*        NX(2)=0
00135     29*        NX(3)=0
00136     30*        LX=0
00137     31*        LGFS=0
00140     32*        IFLAG=0
00141     33*        DO 9 I=1,32
00144     34*      9 BUFF(I)=0
00146     35*        KUOPFC=IHDR(2)
00147     36*        KUDIST=JHDR(2)
00150     37*       10 IF (LOAD(LP).EQ.0) GO TO 18
00152     38*        ISKIP=0
00153     39*        LGF=FLD(2,4,LOAD(LP))
00154     40*        CX=0
00155     41*        LPN=LP+LGF*32
00156     42*        IF (LGF.EQ.0) LPN=LPN+32
00160     43*        LAST=LPN-1
00161     44*        IEND=IF+255
00162     45*        IF (LAST.GT.IEND) LAST=IEND
```

```
00164     46*          DO 11 I=LP,LAST
00167     47*          CX=CX+1
00170     48*          IF (CX.LE.500) GO TO 11
00172     49*          WRITE (6,102) (CASE(C),C=1,4)
00200     50*      102 FORMAT (1H ,'CASE DIMENSION EXCEEDED FOR ',4A6)
00201     51*          STOP
00202     52*       11 CASE(CX)=LOAD(I)
00204     53*          IF (LPN.LT.LMT) GO TO 16
00206     54*          IF (NPREC.NE.IPREC) GO TO 12
00210     55*          A3A=FLD(0,6,LOAD(LP+2))
00211     56*          IF (A3A.NE.CODE) GO TO 20
00213     57*          IFLAG=1
00214     58*          GO TO 17
00215     59*       20 IF (LX.EQ.0) GO TO 27
00217     60*          IHDR(10)=(LGFS+8)/8
00220     61*          DO 21 I=1,32
00223     62*          DX=DP+I-1
00224     63*          IF (I.LE.14) DUMP(DX)=IHDR(I)
00226     64*       21 IF (I.GT.14) DUMP(DX)=0
00231     65*          DP=DP+32
00232     66*          REWIND 39
00233     67*          DO 24 J=1,LX
00236     68*          DX=LONG(J)
00237     69*          READ (39) (DATA(D),D=1,DX)
00245     70*          DPN=DP+DX
00246     71*          LAST=DPN-1
00247     72*          IEND=NF+255
00250     73*          IF (LAST.GT.IEND) LAST=IEND
00252     74*          DX=0
00253     75*          DO 22 I=DP,LAST
00256     76*          DX=DX+1
00257     77*       22 DUMP(I)=DATA(DX)
00261     78*          IF (DPN.LT.LIM) GO TO 24
00263     79*          IF (J.LE.15.OR.J.GE.(LX-7)) WRITE (6,100) (DUMP(D),D=NF,IEND)
00272     80*          CALL WRITE
00273     81*          DPN=DPN-CF
00274     82*          IF (DPN.EQ.NF) GO TO 24
00276     83*          LAST=DPN-1
00277     84*          DO 23 I=NF,LAST
00302     85*          DX=DX+1
00303     86*       23 DUMP(I)=DATA(DX)
00305     87*       24 DP=DPN
00307     88*          IF (DP.EQ.NF) GO TO 26
00311     89*          IEND=NF+255
00312     90*          DO 25 I=DP,IEND
00315     91*       25 DUMP(I)=0
00317     92*          WRITE (6,100) (DUMP(D),D=NF,IEND)
00325     93*          CALL WRITE
00326     94*          DP=NF
00327     95*       26 LX=0
00330     96*          LGFS=0
00331     97*          REWIND 39
00332     98*       27 KUOPEC=IHDR(2)
00333     99*          NPREC=0
00334    100*       12 IF (MPREC.NE.JPREC) GO TO 13
00336    101*       30 I=1
00337    102*      301 CALL SOPEN2 ($31,$33,320,15,3,1,30,0,0,1,4,7,6,0,0,2,3,31,12,0,0,
00337    103*        1      3,14,13,36,0,0,4,99999)
```

```
00340   104*      31 IUNIT=I+8
00341   105*         REWIND IUNIT
00342   106*         IX=NX(I)
00343   107*         DO 32 J=1,IX
00346   108*         CX=NUMB(I,J)
00347   109*         READ (IUNIT) (CASE(C),C=1,CX)
00355   110*      32 CALL SRREL (CASE,CX)
00357   111*         CALL SSORT
00360   112*      33 NX(I)=0
00361   113*         NLGF(I)=0
00362   114*         CALL SRRET (CASE,CX,$40)
00363   115*         NUCASE=FLD(0,30,CASE(3))
00364   116*         REWIND IUNIT
00365   117*         GO TO 35
00366   118*      34 CALL SRRET (CASE,CX,$40)
00367   119*         NUCASE=FLD(0,30,CASE(3))
00370   120*         IF (NUCASE.EQ.KUCASE) GO TO 37
00372   121*         NX(I)=NX(I)+1
00373   122*         J=NX(I)
00374   123*         IF (J.LE.4000) GO TO 342
00376   124*         WRITE (6,341) I
00401   125*     341 FORMAT (1H ,'MULTI-UNIT CASE DIMENSION EXCEEDED FOR YEAR ',I1)
00402   126*         STOP
00403   127*     342 NUMB(I,J)=BX
00404   128*         IF (BX.LE.1000) GO TO 344
00406   129*         WRITE (6,343) BUFF(3)
00411   130*     343 FORMAT (1H ,'MULTI-UNIT CASE DIMENSION EXCEEDED FOR CASE ',A5)
00412   131*         STOP
00413   132*     344 NLGF(I)=NLGF(I)+BUFF(1)
00414   133*         WRITE (IUNIT) (BUFF(B),B=1,BX)
00422   134*      35 BUFF(1)=1
00423   135*         BUFF(2)=0
00424   136*         DO 36 J=3,13
00427   137*      36 BUFF(J)=CASE(J)
00431   138*         FLD(0,12,BUFF(14))=FLD(0,12,CASE(14))
00432   139*         KUCASE=NUCASE
00433   140*         BX=32
00434   141*      37 BUFF(1)=BUFF(1)+FLD(2,4,CASE(1))
00435   142*         IF (FLD(2,4,CASE(1)).EQ.0) BUFF(1)=BUFF(1)+1
00437   143*         BUFF(2)=BUFF(2)+1
00440   144*         DO 38 J=3,13
00443   145*      38 CASE(J)=0
00445   146*         FLD(0,12,CASE(14))=0
00446   147*         DO 39 J=1,CX
00451   148*         BX=BX+1
00452   149*      39 BUFF(BX)=CASE(J)
00454   150*         GO TO 34
00455   151*      40 NX(I)=NX(I)+1
00456   152*         J=NX(I)
00457   153*         NUMB(I,J)=BX
00460   154*         NLGF(I)=NLGF(I)+BUFF(1)
00461   155*         WRITE (IUNIT) (BUFF(B),B=1,BX)
00467   156*         I=I+1
00470   157*         IF (I.LE.3) GO TO 301
00472   158*         DO 41 I=1,9
00475   159*         DX=DP+I-1
00476   160*      41 DUMP(DX)=MUNDR(I)
00500   161*         DUMP(DP+9)=(NLGF(1)+NLGF(2)+NLGF(3)+8)/8
```

```
00501    162*          DUMP(DP+10)=NX(1)+NX(2)+NX(3)
00502    163*          DUMP(DP+11)=NX(1)
00503    164*          DUMP(DP+12)=NX(2)
00504    165*          DUMP(DP+13)=NX(3)
00505    166*          DO 410 I=15,32
00510    167*          DX=DP+I-1
00511    168*      410 DUMP(DX)=0
00513    169*          DP=DP+32
00514    170*          I=1
00515    171*          REWIND 39
00516    172*      411 CALL SOPEN3 ($42,$44,1000,47,4,7,6,0,0,1,3,31,12,0,0,2,46,13,36,0,
00516    173*         1      0,3,99999)
00517    174*       42 IUNIT=I+8
00520    175*          REWIND IUNIT
00521    176*          IQUIT=IOREX+20
00522    177*          IX=NX(I)
00523    178*          DO 43 J=1,IX
00526    179*          BX=NUMB(I,J)
00527    180*          READ (IUNIT) (BUFF(B),B=1,BX)
00535    181*       43 CALL SRREL (BUFF,BX)
00537    182*          CALL SSORT
00540    183*       44 CALL SRRET (BUFF,BX,$48)
00541    184*          BY=BX
00542    185*          BX=0
00543    186*      441 DPN=DP+32
00544    187*          LAST=DPN-1
00545    188*          IEND=NF+255
00546    189*          IF (LAST.GT.IEND) LAST=IEND
00550    190*          DO 45 J=DP,LAST
00553    191*          BX=BX+1
00554    192*       45 DUMP(J)=BUFF(BX)
00556    193*          IF (DPN.LT.LIM) GO TO 47
00560    194*          IF (IOREX.LE.IQUIT) WRITE (6,100) (DUMP(D),D=NF,IEND)
00567    195*          CALL WRITE
00570    196*          DPN=DPN-CF
00571    197*          IF (DPN.EQ.NF) GO TO 47
00573    198*          LAST=DPN-1
00574    199*          DO 46 J=NF,LAST
00577    200*          BX=BX+1
00600    201*       46 DUMP(I)=BUFF(BX)
00602    202*       47 DP=DPN
00603    203*          IF (BX.NE.BY) GO TO 441
00605    204*          GO TO 44
00606    205*       48 I=I+1
00607    206*          IF (I.LE.3) GO TO 411
00611    207*          IF (DP.EQ.NF) GO TO 50
00613    208*          IEND=NF+255
00614    209*          DO 49 J=DP,IEND
00617    210*       49 DUMP(J)=0
00621    211*          WRITE (6,100) (DUMP(D),D=NF,IEND)
00627    212*          CALL WRITE
00630    213*       50 IQT=1
00631    214*          CALL WRITE
00632    215*          CALL NTRAN (8,9)
00633    216*          STOP
00633    217*    C     MPREC=0
00634    218*       13 CALL READ ($30)
00635    219*          LPN=LPN-KF
```

```
00636   220*          IF (LPN.NE.IF) GO TO 14
00640   221*          IF (NPREC.EQ.1) LPN=LPN+32
00642   222*          IF (ISKIP.EQ.1.OR.IFLAG.EQ.1) GO TO 19
00644   223*          GO TO 16
00645   224*       14 LAST=LPN-1
00646   225*          DO 15 I=IF,LAST
00651   226*          CX=CX+1
00652   227*       15 CASE(CX)=LOAD(I)
00654   228*       16 A3A=FLD(0,6,LOAD(LP+2))
00655   229*          IF (A3A.EQ.CODE) GO TO 17
00657   230*          I=FLD(2,4,LOAD(LP+3))
00660   231*          J=FLD(32,4,LOAD(LP+2))
00661   232*          A4A=I+J*10
00662   233*          A4B=FLD(8,4,LOAD(LP+3))
00663   234*          I=2
00664   235*          IF (A4B.EQ.6.OR.(A4B.EQ.7.AND.A4A.LT.7)) I=1
00666   236*          IF (A4B.EQ.9.OR.(A4B.EQ.8.AND.A4A.GE.7)) I=3
00670   237*          IHDR(11)=IHDR(11)-1
00671   238*          IHDR(I+11)=IHDR(I+11)-1
00672   239*          NX(I)=NX(I)+1
00673   240*          J=NX(I)
00674   241*          NUMB(I,J)=CX
00675   242*          IUNIT=I+8
00676   243*          WRITE (IUNIT) (CASE(C),C=1,CX)
00704   244*      100 FORMAT (2(16(1X,A6)/))
00705   245*      101 FORMAT (1H+,T120,'UNIT=',I6)
00706   246*          GO TO 19
00707   247*       17 LX=LX+1
00710   248*          IF (LX.LE.4000) GO TO 172
00712   249*          WRITE (6,171) IHDR(2)
00715   250*      171 FORMAT (1H ,'UNIT CASE DIMENSION EXCEEDED FOR OPFAC ',A5)
00716   251*          STOP
00717   252*      172 LONG(LX)=CX
00720   253*          LGFS=LGFS+LGF
00721   254*          IF (LGF.EQ.0) LGFS=LGFS+1
00723   255*          WRITE (39) (CASE(C),C=1,CX)
00731   256*          IF (IFLAG.EQ.1) GO TO 20
00733   257*          GO TO 19
00734   258*       18 ISKIP=1
00735   259*          LPN=LP+32
00736   260*          IF (LPN.LT.LMT) GO TO 19
00740   261*          IF (NPREC.EQ.IPREC) GO TO 20
00742   262*          IF (MPREC.EQ.JPREC) GO TO 30
00744   263*          CALL READ ($30)
00745   264*          LPN=LPN-KF
00746   265*          IF (NPREC.EQ.1) LPN=LPN+32
00750   266*       19 LP=LPN
00751   267*          IFLAG=0
00752   268*          GO TO 10
00753   269*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    1 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    1 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    1 SEC.
PHASE 6 TIME =    0 SEC.
```

```
@IT FOR READ,READ                                                                    03 JUN 71          22:51:15.053
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F501RP
THIS COMPILATION WAS DONE ON 03 JUN 71 AT 22:51:15


   SUBROUTINE READ       ENTRY POINT 000326

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000337
         0000    *DATA    000045
         0002    *BLANK   003427
         0003    BREAD    000011
         0004    BWRITE   000011


   EXTERNAL REFERENCES (BLOCK, NAME)

         0005    NTRAN
         0006    CLOCKS
         0007    NWDU$
         0010    NIO2$
         0011    NSTOP$
         0012    NERR4$
         0013    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000007 1L          0000    000012 100F        0001    000233 11L         0001    000241 12L         0001    000147 154G
    0001    000217 177G        0001    000021 2L          0001    000034 3L          0001    000062 4L          0001    000115 5L
    0001    000121 6L          0001    000204 8L          0001    000211 9L          0004 I 000004 CF          0000 I 000004 DERR
    0004 I 000006 DP           0004 I 000005 DPN          0002 I 001000 DUMP         0002 R 002000 E            0000 I 000006 I
    0003 I 000001 IF           0002 I 002005 IHDR         0003 I 000003 INREX        0004 I 000003 IOREX        0002 I 003423 IPREC
    0002 I 002003 IQT          0003 I 000000 IS           0000 I 000001 ISKIP        0003 I 000007 ISTAT        0000 I 000002 IT
    0002 I 002004 IUNIT        0000 I 000000 IX           0000 I 000003 JF           0002 I 002023 JHDR         0002 I 003424 JPREC
    0000 I 000007 JX           0003 I 000004 KF           0004 I 000002 LIM          0003 I 000002 LMT          0002 I 000000 LOAD
    0003 I 000006 LP           0003 I 000005 LPN          0000 I 000010 LX           0002 I 003426 MPREC        0000 I 000005 NEXT
    0004 I 000001 NF           0000 I 000011 NPR          0002 I 003425 NPREC        0004 I 000000 NS           0004 I 000007 NSTAT
    0002 R 002001 X0           0002 R 002002 X1


00101      1*          SUBROUTINE READ (S)
00103      2*          IMPLICIT INTEGER (A,B,C,D)
00104      3*          COMMON LOAD(512),DUMP(512),E,X0,X1,IQT,IUNIT,IHDR(14),JHDR(768),
00104      4*         I      IPREC,JPREC,NPREC,MPREC
00105      5*          COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106      6*          COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107      7*          IF (MPREC.GT.0) GO TO 1
00111      8*          IX=1
00112      9*          ISKIP=0
00113     10*        1 IT=3-IS
00114     11*          JF=258-IF
00115     12*          KF=IF+IF-2
```

```
00116     13*        2 IF (ISTAT(IT)+1),2,6
00121     14*          IF (ISTAT(IT)+2),5,6
00124     15*          DERR=1
00125     16*        3 CALL NTRAN (7,22)
00126     17*          CALL NTRAN (7,7,-1)
00127     18*          CALL NTRAN (7,2,256,LOAD(JF),ISTAT(IT))
00130     19*        4 IF (ISTAT(IT)+1),4,6
00133     20*          IF (ISTAT(IT)+2),5,6
00136     21*          DERR=DERR+1
00137     22*          IF (DERR.LT.4) GO TO 3
00141     23*          NEXT=INREX+1
00142     24*          WRITE (6,100) NEXT
00145     25*          STOP
00146     26*        5 RETURN 1
00147     27*        6 IF (IQT.EQ.1) RETURN
00151     28*          IF (MPREC.GT.0.OR.ISKIP.EQ.1) GO TO 9
00153     29*          DO 7 I=1,256
00156     30*          JX=IX+I-1
00157     31*          LX=JF+I-1
00160     32*        7 JHDR(JX)=LOAD(LX)
00162     33*          IF (IX.EQ.1) NPR=JHDR(1)
00164     34*          NPR=NPR-1
00165     35*          IF (NPR.EQ.0) GO TO 8
00167     36*          IX=IX+256
00170     37*          GO TO 12
00171     38*        8 JPREC=JHDR(3)
00172     39*          ISKIP=1
00173     40*          GO TO 12
00174     41*        9 IF (NPREC.GT.0) GO TO 11
00176     42*          DO 10 I=1,14
00201     43*          LX=JF+I-1
00202     44*       10 IHDR(I)=LOAD(LX)
00204     45*          IPREC=IHDR(10)
00205     46*       11 NPREC=NPREC+1
00206     47*          MPREC=MPREC+1
00207     48*       12 INREX=INREX+1
00210     49*          CALL NTRAN (7,2,256,LOAD(IF),ISTAT(IS))
00211     50*          IS=IT
00212     51*          IF=JF
00213     52*          LMT=708-LMT
00214     53*          CALL CLOCKS(X1)
00215     54*          IF (X1.GE.X0) RETURN 1
00217     55*          IF (MPREC.EQ.0) GO TO 1
00221     56*          RETURN
00222     57*      100 FORMAT (1H ,'UNABLE TO READ RECORD ',I6)
00223     58*          END

       END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    0 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    0 SEC.
PHASE 6 TIME =    1 SEC.


TOTAL COMPILATION TIME =    1  SEC
```

```
@IT FOR WRITE,WRITE                                                                    03 JUN 71          22:51:16.268
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 FS01WP
THIS COMPILATION WAS DONE ON 03 JUN 71 AT 22:51:16


   SUBROUTINE WRITE      ENTRY POINT 000170

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000174
         0000    *DATA    000045
         0002    *BLANK   003427
         0003    BREAD    000011
         0004    BWRITE   000011


   EXTERNAL REFERENCES (BLOCK, NAME)

         0005    NTRAN
         0006    NWDU$
         0007    NIO2$
         0010    NSTOP$
         0011    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000012 1L          0000    000004 100F        0000    000012 101F        0001    000025 2L          0001    000053 3L
    0001    000106 4L          0001    000115 5L          0004 I 000004 CF          0000 I 000002 DERR        0004 I 000006 DP
    0004 I 000005 DPN          0002 I 001000 DUMP         0002 R 002000 E           0003 I 000001 IF          0002 I 002005 IHDR
    0003 I 000003 INREX        0004 I 000003 IOREX        0002 I 003423 IPREC       0002 I 002003 IQT         0003 I 000000 IS
    0003 I 000007 ISTAT        0002 I 002004 IUNIT        0002 I 002023 JHDR        0002 I 003424 JPREC       0003 I 000004 KF
    0004 I 000002 LIM          0003 I 000002 LMT          0002 I 000000 LOAD        0003 I 000006 LP          0003 I 000005 LPN
    0000 I 000001 MF           0002 I 003426 MPREC        0000 I 000003 NEXT        0004 I 000001 NF          0002 I 003425 NPREC
    0004 I 000000 NS           0004 I 000007 NSTAT        0000 I 000000 NT          0002 R 002001 X0          0002 R 002002 X1


00101      1*          SUBROUTINE WRITE
00103      2*          IMPLICIT INTEGER (A,B,C,D)
00104      3*          COMMON LOAD(512),DUMP(512),E,X0,X1,IQT,IUNIT,IHDR(14),JHDR(768),
00104      4*        1     IPREC,JPREC,NPREC,MPREC
00105      5*          COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106      6*          COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107      7*          NT=3-NS
00110      8*          MF=258-NF
00111      9*          CF=NF+NF-2
00112     10*        1 IF (NSTAT(NT)+1),1,5
00115     11*          IF (NSTAT(NT)+2),4,5
00120     12*          DERR=1
00121     13*        2 CALL NTRAN (8,22)
00122     14*          CALL NTRAN (8,7,-1)
00123     15*          CALL NTRAN (8,1,256,DUMP(MF),NSTAT(NT))
00124     16*        3 IF (NSTAT(NT)+1),3,5
00127     17*          IF (NSTAT(NT)+2),4,5
```

```
00132     18*          DERR=DERR+1
00133     19*          IF (DERR.LT.4) GO TO 2
00135     20*          NEXT=IOREX+1
00136     21*          WRITE (6,100) NEXT
00141     22*          STOP
00142     23*        4 WRITE (6,101)
00144     24*          STOP
00145     25*        5 IF (IQT.EQ.1) RETURN
00147     26*          IOREX=IOREX+1
00150     27*          CALL NTRAN (8,1,256,DUMP(NF),NSTAT(NS))
00151     28*          NS=NT
00152     29*          NF=MF
00153     30*          LIM=708-LIM
00154     31*          RETURN
00155     32*      100 FORMAT (1H ,'UNABLE TO WRITE RECORD ',I6)
00156     33*      101 FORMAT (1H ,'ATTEMPT TO WRITE PAST END OF TAPE')
00157     34*          END

       END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    0 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    0 SEC.
PHASE 6 TIME =    1 SEC.

TOTAL COMPILATION TIME =    1  SEC
```

MUC130

Like MUTAPE, the source deck must be changed in this program to correspond to the data being processed. In the main program, card number 7 must state information for the C-130 station under consideration. The district number goes in columns 20 and 21, OPFAC number in columns 25 - 29, and OPFAC name in column 36 of card 7 through column 11 of card 8. In the listing given, the C-130 station is San Francisco, OPFAC number 20180 in district 12.

In Subroutine READ, the OPFAC number must be given in card 7, columns 21 through 25.

```
@IT FOR MUC130,MUC130                                                                    06 AUG 71          04:37:45.157
UNIVAC 1108 FORTRAN V LEVEL   9206 001, FS0192
THIS COMPILATION WAS DONE 01 05 AUG 71 AT 04:37:45


MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE   001007
      0000   *DATA   003026
      0002   *BLANK  003427
      0003   BREAD   000011
      0004   BWRITE  000011


EXTERNAL REFERENCES (BLOCK, NAME)

      0005   CLOCKS
      0006   NTRAN
      0007   READ
      0010   WRITE
      0011   NWB0$
      0012   NIO1$
      0013   NIO2$
      0014   NRE $
      0015   NRBU$
      0016   NWDU$
      0017   NSTOP$


STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000114 13L        0000   003005 100F       0001   000215 13L        0001   000217 14L        0001   000064 145G
 0001   000226 15L        0001   000243 16L        0001   000247 17L        0001   000146 172G       0001   000310 19L
 0001   000312 20L        0001   000330 21L        0001   000206 211G       0001   000344 22L        0001   000472 23L
 0001   000505 242G       0001   000477 30L        0001   000466 306G       0001   000520 31L        0001   000523 32L
 0001   000553 341G       0001   000613 354G       0001   000624 361G       0001   000656 374G       0001   000526 40L
 0001   000701 415G       0001   000724 420G       0001   000754 433G       0001   000733 44L        0001   000765 441G
 0001   000773 50L        0001   000030 6L         0001   000062 7L         0001   000111 9L         0000 I 003001 A4A
 0000 I 003002 A4B        0004 I 000004 CF         0000 I 002734 CHDR       0000 I 002772 C130       0000 I 003003 D
 0000 I 001750 DATA       0004 I 000006 DP         0004 I 000005 DPN        0002 I 001000 DUMP       0000 I 002763 DX
 0000 I 002775 D1         0002 R 002000 E          0000 I 002761 I          0000 I 002766 IEND       0003 I 000001 IF
 0002 I 002003 IHDR       0003 I 000003 INREX      0004 I 000003 IOREX      0002 I 003423 IPREC      0002 I 003422 IOT
 0000 I 002760 IQUIT      0003 I 000000 IS         0003 I 000007 ISTAT      0002 I 003421 IUMIT      0000 I 002767 IX
 0000 I 002770 IY         0000 I 002774 IZ         0000 I 002755 I3         0000 I 002773 J          0002 I 002021 JHDR
 0002 I 003424 JPREC      0000 I 003000 JX         0000 I 002776 JY         0000 I 002777 JZ         0000 I 002771 KD3
 0003 I 000004 KF         0000 I 002765 LAST       0000 I 002764 LGF        0000 I 002757 LGFS       0004 I 000002 LIM
 0003 I 000002 LMT        0002 I 000000 LOAD       0003 I 000006 LP         0003 I 000005 LPN        0000 I 002762 LX
 0002 I 003426 MPREC      0000 I 002752 MX         0000 I 003004 N          0004 I 000001 NF         0002 I 003425 NPREC
 0004 I 000000 NS         0004 I 000007 NSTAT      0000 I 000000 NWDS       0000 I 002756 NX         0002 R 002001 X0
 0002 R 002002 X1
```

```
00101     1*        IMPLICIT INTEGER (A-B,C-Z)
00103     2*        COMMON LOAD(513),DUMP(512),E,X0,X1,IHDR(14),JHDR(768),IQUIT,IOT,
00103     3*       1      IPREC,JPREC,NPREC,MPREC
00104     4*        COMMON /BREAD/ IS,IF,LMT,INREX,E,LPN,LP,ISTAT(2)
00105     5*        COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00106     6*        DIMENSION NAMS(1000),DATA(500),CHDR(14),MX(3)
00107     7*        DATA CHDR /' 12   28180        SA  FRANCISCO C130                ',
00107     8*       1      '                                                          '/
00111     9*        E=180.0
00112    10*        CALL CLOCKS(X0)
00113    11*        X0=X0+E
00114    12*        IQT=0
00115    13*        NS=1
00116    14*        NF=1
00117    15*        LIM=226
00120    16*        IOREX=0
00121    17*        DP=NF
00122    18*        MX(1)=0
00123    19*        MX(2)=0
00124    20*        MX(3)=0
00125    21*        NX=0
00126    22*        LGFS=0
00127    23*        IQUIT=0
00130    24*        IUNIT=7
00131    25*      6 CALL NTRAN (IUNIT,2,256,LOAD(1),ISTAT(1))
00132    26*        IS=2
00133    27*        IF=257
00134    28*        LMT=482
00135    29*        INREX=0
00136    30*        NPREC=0
00137    31*        MPREC=0
00140    32*        CALL READ ($50)
00141    33*        LP=IF+32
00142    34*        IF (IUNIT.EQ.9) GO TO 10
00144    35*      7 DO 8 I=1,256
00147    36*        LX=IF+I-1
00150    37*        DX=NF+I-1
00151    38*      8 DUMP(DX)=LOAD(LX)
00153    39*        CALL WRITE
00154    40*        CALL READ ($9)
00155    41*        GO TO 7
00156    42*      9 IUNIT=9
00157    43*        GO TO 6
00160    44*     10 LGF=FLD(2,4,LOAD(LP))
00161    45*        IF (LGF.EQ.0) GO TO 30
00163    46*        DX=0
00164    47*        LPN=LP+LGF*32
00165    48*        LAST=LPN-1
00166    49*        IEND=IF+255
00167    50*        IF (LAST.GT.IEND) LAST=IEND
00171    51*        DO 11 I=LP,LAST
00174    52*        DX=DX+1
00175    53*     11 DATA(DX)=LOAD(I)
00177    54*        IF (LPN.LT.LMT) GO TO 13
00201    55*        IF (NPREC.EQ.IPREC) GO TO 32
00203    56*        CALL READ ($50)
00204    57*        LPN=LPN-KF
00205    58*        IF (LPN.EQ.IF) GO TO 13
```

```
00207    59*          LAST=LP+1
00210    60*          DO 12 I=1F,LAS
00213    61*          JX=JX+1
00214    62*       1  DATA(JX)=LOAD( )
00215    63*       1  LP=LPN
00217    64*       1  IX=1
00220    65*          IY=24
00221    66*          KD3=0
00222    67*          C130=0
00223    68*          GO TO 17
00224    69*       1  I=FLD(8,4,DATA(IX))
00225    70*          J=FLD(2,4,DATA(IX))
00226    71*          KD3=I+J*10-1
00227    72*       1  IY=IX
00230    73*          IZ=IX+18
00231    74*       17 I=FLD(20,4,DATA(IY))
00232    75*          J=FLD(14,4,DATA(IY))
00233    76*          D1=I+J*10
00234    77*          IF (D1.EQ.60) GO TO 19
00236    78*          FLD(12,24,DATA(IY))=0
00237    79*          JY=IY+1
00240    80*          JZ=IY+8
00241    81*          DO 18 JX=JY,JZ
00244    82*       1  DATA(JX)=0
00246    83*          GO TO 20
00247    84*       19 C130=1
00250    85*       2  IF (KD3.EQ.0) GO TO 21
00252    86*          KD3=KD3-1
00253    87*          IF (KD3.EQ.0) GO TO 22
00255    88*          IY=IY+9
00256    89*          IF (IY.LE.IZ) GO TO 17
00260    90*       21 IX=IX+32
00261    91*          IF (KD3.GT.0) GO TO 16
00263    92*          IF (LGF.GT.1) GO TO 15
00265    93*       22 IF (C130.EQ.0) GO TO 23
00267    94*          I=FLD(2,4,DATA(4))
00270    95*          J=FLD(32,4,DATA(3))
00271    96*          A4A=I+J*10
00272    97*          A4B=FLD(8,4,DATA(4))
00273    98*          I=2
00274    99*          IF (A4B.EQ.6.OR.(A4B.EQ.7.AND.A4A.LT.7)) I=1
00276   100*          IF (A4B.EQ.9.OR.(A4B.EQ.8.AND.A4A.GE.7)) I=3
00300   101*          MX(I)=MX(I)+1
00301   102*          NX=NX+1
00302   103*          NWDS(NX)=JX
00303   104*          LGFS=LGFS+LGF
00304   105*          WRITE (36) (DATA(D),D=1,JX)
00312   106*       23 IF (IQUIT.EQ.1) GO TO 40
00314   107*          GO TO 10
00315   108*       30 LPN=LP+32
00316   109*          IF (LPN.LT.LMT) GO TO 31
00320   110*          IF (NPREC.EQ.IPREC) GO TO 40
00322   111*          CALL READ ($50)
00323   112*          LPN=LP+1-KF
00324   113*       31 LP=LPN
00325   114*          GO TO 10
00326   115*       32 IQUIT=1
00327   116*          GO TO 14
```

```
00330   117*        4  IF (NX.EQ.0) GO TO 50
00332   118*           DP=NF
00333   119*           CHDR(10)=(LGFS+8)/3
00334   120*           CHDR(11)=NX
00335   121*           CHDR(12)=NX(1)
00336   122*           CHDR(13)=NX(2)
00337   123*           CHDR(14)=NX(3)
00340   124*           DO 41 I=1,32
00343   125*           DX=DP+I-1
00344   126*           IF (I.LE.14) DUMP(DX)=CHDR(I)
00346   127*        41 IF (I.GT.14) DUMP(DX)=0
00351   128*           DP=DP+32
00352   129*           REWIND 36
00353   130*           DO 44 N=1,NX
00356   131*           DX=NWDS(N)
00357   132*           READ (36) (DATA(D),D=1,DX)
00365   133*           DPN=DP+DX
00366   134*           LAST=DPN-1
00367   135*           IEND=NF+255
00370   136*           IF (LAST.GT.IEND) LAST=IEND
00372   137*           DX=0
00373   138*           DO 42 I=DP,LAST
00376   139*           DX=DX+1
00377   140*        42 DUMP(I)=DATA(DX)
00401   141*           IF (DPN.LT.LIM) GO TO 44
00403   142*           WRITE (6,100) (DUMP(D),D=NF,IEND)
00411   143*       100 FORMAT (2(16(1X,A6)/))
00412   144*           CALL WRITE
00413   145*           DPN=DPN-CF
00414   146*           IF (DPN.EQ.NF) GO TO 44
00416   147*           LAST=DPN-1
00417   148*           DO 43 I=NF,LAST
00422   149*           DX=DX+1
00423   150*        43 DUMP(I)=DATA(DX)
00425   151*        44 DP=DPN
00427   152*           IF (DP.EQ.NF) GO TO 50
00431   153*           IEND=NF+255
00432   154*           DO 45 I=DP,IEND
00435   155*        45 DUMP(I)=0
00437   156*           WRITE (6,100) (DUMP(D),D=NF,IEND)
00445   157*           CALL WRITE
00446   158*        50 IQT=1
00447   159*           CALL WRITE
00450   160*           CALL NTRAN (8,0)
00451   161*           STOP
00452   162*           END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   1 SEC.


TOTAL COMPILATION TIME =   2  SEC
```

```
SUBROUTINE READ        ENTRY POINT 000400

STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE   000411
      0000   *DATA   000054
      0002   *BLANK  003427
      0003   BREAD   000011
      0004   BWRITE  000011


EXTERNAL REFERENCES (BLOCK, NAME)

      0005   NTRAN
      0006   CLOCKS
      0007   NWOO$
      0010   NIO2$
      0011   NSTOP$
      0012   NIO1$
      0013   NERR4$
      0014   NERR3$


STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000017 1L          0000   000017 100F        0000   000014 101F        0001   000276 11L         0001   000303 12L
 0001   000162 160G        0001   000031 2L          0001   000232 200G        0001   000245 211G        0001   000044 3L
 0001   000072 4L          0001   000125 5L          0001   000131 6L          0001   000220 8L          0001   000237 9L
 0004 I 000004 CF          0000 I 000005 DERR        0004 I 000006 DP          0004 I 000005 DPN         0002 I 001000 DUMP
 0002 R 002000 E           0000 I 000007 I           0003 I 000001 IF          0002 I 002003 IHDR        0003 I 000003 INREX
 0000 I 000000 IOPFAC      0004 I 000003 IOREX       0002 I 003423 IPREC       0002 I 003422 IOT         0003 I 000000 IS
 0000 I 000002 ISKIP       0003 I 000007 ISTAT       0000 I 000003 IT          0002 I 003421 IUNIT       0000 I 000001 IX
 0000 I 000013 J           0000 I 000004 JF          0002 I 002021 JHDR        0002 I 003424 JPREC       0000 I 000010 JX
 0003 I 000004 KF          0004 I 000002 LIM         0003 I 000002 LMT         0002 I 000000 LOAD        0003 I 000006 LP
 0005 I 000005 LPN         0000 I 000011 LX          0002 I 003426 MPREC       0000 I 000006 NEXT        0004 I 000001 NF
 0000 I 000012 NPR         0002 I 003425 NPREC       0004 I 000000 NS          0004 I 000007 NSTAT       0002 R 002001 X0
 0002 R 002002 X1
```

```
00101     1*          SUBROUTINE READ ($)
00103     2*          IMPLICIT INTEGER (A,B,C,D)
00104     3*          COMMON LOAD(512),DUMP(512),E,X0,X1,IHDR(14),JHDR(768),IUNIT,IOT,
00104     4*         1     IPREC,JPREC,NPREC,MPREC
00105     5*          COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106     6*          COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107     7*          DATA IOPFAC /'20180 '/
00111     8*          IF (IUNIT.EQ.7.OR.MPREC.GT.0) GO TO 1
00113     9*          IX=1
00114    10*          ISKIP=0
```

```
00115    11*          IT=3-IS
00116    12*          JF=256-IF
00117    13*          KF=IF+IF-2
00120    14*          IF (ISTAT(IT)+1),2,5
00123    15*          IF (ISTAT(IT)+2),5,5
00126    16*          DERR=1
00127    17*        2 CALL NTRAN (IUNIT,22)
00130    18*          CALL NTRAN (IUNIT,7,-1)
00131    19*          CALL NTRAN (IUNIT,2,256,LOAD(JF),ISTAT(IT))
00132    20*        4 IF (ISTAT(IT)+1),4,6
00135    21*          IF (ISTAT(IT)+2),5,6
00140    22*          DERR=DERR+1
00141    23*          IF (DERR.LT.4) GO TO 3
00143    24*          NEXT=INREX+1
00144    25*          WRITE (6,100) NEXT
00147    26*          STOP
00150    27*        6 RETURN 1
00151    28*        5 IF (IQT.EQ.1) RETURN
00153    29*          IF (IUNIT.EQ.7) GO TO 12
00155    30*          IF (MPREC.GT.0.OR.ISKIP.EQ.1) GO TO 9
00157    31*          DO 7 I=1,256
00162    32*          JX=IX+I-1
00163    33*          LX=JF+I-1
00164    34*        7 JHDR(JX)=LOAD(LX)
00166    35*          IF (IX.EQ.1) NPR=JHDR(1)
00170    36*          NPR=NPR-1
00171    37*          IF (NPR.EQ.0) GO TO 8
00173    38*          IX=IX+256
00174    39*          GO TO 12
00175    40*        8 ISKIP=1
00176    41*          WRITE (6,101) (JHDR(J),J=1,JX)
00204    42*      101 FORMAT (2(16(1X,A6)/))
00205    43*          GO TO 12
00206    44*        9 IF (NPREC.GT.0) GO TO 11
00210    45*          DO 10 I=1,14
00213    46*          LX=JF+I-1
00214    47*       10 IHDR(I)=LOAD(LX)
00216    48*          IPREC=IHDR(10)
00217    49*          IF (IHDR(2).EQ.IOPFAC) GO TO 11
00221    50*          CALL NTRAN (IUNIT,7,IPREC-1)
00222    51*          GO TO 12
00223    52*       11 NPREC=NPREC+1
00224    53*          MPREC=MPREC+1
00225    54*       12 INREX=INREX+1
00226    55*          CALL NTRAN (IUNIT,2,256,LOAD(IF),ISTAT(IS))
00227    56*          IS=IT
00230    57*          IF=JF
00231    58*          LMT=708-LMT
00232    59*          CALL CLOCKS(X1)
00233    60*          IF (X1.GE.X0) RETURN 1
00235    61*          IF (IUNIT.EQ.9.AND.MPREC.EQ.0) GO TO 1
00237    62*          RETURN
00240    63*      100 FORMAT (1H ,'UNABLE TO READ RECORD ',I6)
00241    64*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
```

```
UNIVAC 1108 FORTRAN V LEVEL   2206  01 X 750182
THIS COMPILATION WAS DONE ON 06 AUG 71 AT 04:37:48


   SUBROUTINE WRITE      ENTRY POINT 000170

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000174
         0000   *DATA   000045
         0002   *BLANK  003427
         0003   BREAD   000011
         0004   BWRITE  000011


   EXTERNAL REFERENCES (BLOCK, NAME)

         0005   NTRAN
         0006   NWD$$
         0007   NIO2$
         0010   NSTOP$
         0011   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000012 1L          0000   000004 100F        0000   000012 101F        0001   000025 2L          0001   000053 3L
    0001   000106 4L          0001   000115 5L          0004 I 000004 CF          0000 I 000002 DERR        0004 I 000006 DP
    0004 I 000005 DPN         0002 I 001000 DUMP        0002 R 002000 E           0003 I 000001 IF          0002 I 002003 IHDR
    0003 I 000003 IOREX       0004 I 000003 IOREX       0002 I 003423 IPREC       0002 I 003422 IQT         0003 I 000000 IS
    0003 I 000007 ISTAT       0002 I 003421 IUNIT       0002 I 002021 JHDR        0002 I 003424 JPREC       0003 I 000004 KF
    0004 I 000002 LIM         0003 I 000002 LMT         0002 I 000000 LOAD        0003 I 000006 LP          0003 I 000005 LPN
    0000 I 000001 MF          0002 I 003426 MPREC       0000 I 000003 NEXT        0004 I 000001 NF          0002 I 003425 NPREC
    0004 I 000000 NS          0004 I 000007 NSTAT       0000 I 000000 NT          0002 R 002001 X0          0002 R 002002 X1


00101     1*          SUBROUTINE WRITE
00103     2*          IMPLICIT INTEGER (A,B,C,D)
00104     3*          COMMON LOAD(512),DUMP(512),E,X0,X1,IHDR(14),JHDR(768),IUNIT,IQT,
00104     4*         1      IPREC,JPREC,NPREC,MPREC
00105     5*          COMMON /BREAD/ IS,IF,LMT,INREX,KF,LPN,LP,ISTAT(2)
00106     6*          COMMON /BWRITE/ NS,NF,LIM,IOREX,CF,DPN,DP,NSTAT(2)
00107     7*          NT=3-NS
00110     8*          MF=258-NF
00111     9*          CF=NF+NF-2
00112    10*        1 IF (NSTAT(NT)+1),1,5
00115    11*          IF (NSTAT(NT)+2),4,5
00120    12*          DERR=1
00121    13*        2 CALL NTRAN (8,22)
00122    14*          CALL NTRAN (8,7,-1)
00123    15*          CALL NTRAN (8,1,256,DUMP(MF),NSTAT(NT))
00124    16*        3 IF (NSTAT(NT)+1),3,5
00127    17*          IF (NSTAT(NT)+2),4,5
```

```
00152    18*          DERR=DERR+1
00153    19*          IF (DERR.LT.4) GO TO 2
00155    20*          NEXT=IOREX+1
00156    21*          WRITE (6,100)  EXT
00141    22*          STOP
00142    23*        2 WRITE (6,101)
00144    24*          STOP
00145    25*        1 IF (IOT.EQ.1) RETURN
00147    26*          IOREX=IOREX+1
00150    27*          CALL NTRAN (8,1,256,DUMP(NF),NSTAT(NS))
00151    28*          NS=NT
00152    29*          NF=MF
00153    30*          LIM=708-LIM
00154    31*          RETURN
00155    32*      100 FORMAT (1H ,'UNABLE TO WRITE RECORD ',I6)
00156    33*      101 FORMAT (1H ,'ATTEMPT TO WRITE PAST END OF TAPE')
00157    34*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   0 SEC.


TOTAL COMPILATION TIME =   1  SEC
```

MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

```
0001   *CODE   003504
0000   *DATA   025313
0002   *BLANK  022543
```

EXTERNAL REFERENCES (BLOCK, NAME)

```
0003   NTRAN
0004   READ
0005   NUCASE
0006   RANF
0007   NEEDS
0010   NRDJ$
0011   NIO2$
0012   NIO1$
0013   NWDJ$
0014   COS
0015   TAN
0016   SIN
0017   ATAN
0020   ASIN
0021   NWBJ$
0022   NSTOP$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0001   000061 1L        0001   000405 10L       0001   002362 1007G     0000   025154 101F      0001   002411 1017G
0000   025157 102F      0000   025161 103F      0000   025172 104F      0001   000300 105L      0000   025175 106F
0000   025180 107F      0001   002553 1070G     0001   000601 11L       0001   002631 1107G     0001   000776 111L
0001   001067 112L      0001   001133 113L      0000   025177 115F      0001   001163 116L      0001   002767 1160G
0001   001172 117L      0001   001175 118L      0001   001263 119L      0001   001355 121L      0001   003123 1212G
0001   001365 122L      0001   003257 1272G     0001   001312 12L       0001   003374 1360G     0001   001371 14L
0001   003435 1403G     0001   003443 1411G     0001   003453 1417G     0001   003466 1427G     0001   000043 151G
0001   000055 160G      0001   001423 17L       0001   000202 2L        0001   001450 20L       0001   003403 200L
0000   025220 201F      0000   025223 203F      0000   025242 205F      0001   001474 21L       0001   001551 24L
0001   001553 25L       0001   002020 26L       0001   001773 260L      0001   001777 261L      0001   002022 27L
0001   002027 28L       0001   002224 29L       0001   000210 3L        0000   025245 300F      0000   025253 302F
0001   000343 323G      0001   000420 323G      0001   002240 35L       0001   002243 351L      0001   000627 353G
0001   001006 377G      0001   000226 4L        0001   001141 424G      0001   001206 450G      0001   001333 477G
0000   025152 498F      0000   025153 499F      0001   000303 5L        0001   002246 50L       0000   025165 500F
0001   002275 511L      0001   002344 512L      0001   002347 52L       0001   002356 520L      0001   001413 526G
0001   002436 53L       0001   002440 54L       0001   002454 55L       0001   001501 557G      0001   002503 56L
0001   001530 557G      0001   002531 60L       0001   001722 640G      0001   001756 652G      0001   002005 666G
0001   000400 7L        0001   002533 70L       0001   002033 703G      0001   002063 713G      0001   002644 72L
0001   002673 73L       0001   002614 74L       0001   002706 75L       0001   002303 763G      0001   002333 773G
0001   002717 78L       0001   002743 80L       0001   003017 81L       0001   003021 90L       0001   003133 92L
```

```
0000    025215 9 F        0001    003316 94L        0000    025206 96F        0000    025211 96E        0001    003136 97L
0000    025215 9 F        0001    003320 99L        0000 I 025070 A           0000 I 025132 A550        0002 I 002313 A1A
0002 I 002475 A13         0002 I 002457 A3          0002 I 002472 A3A         0002 I 022645 A3OLD       0000 I 025151 A4
0002 I 002469 A4A         0002 I 002461 A43         0002 I 013240 B           0000 I 025114 BD          0000 I 001136 BDCASE
0000 I 025133 BLAT        0000 I 025115 BX          0002 I 002467 B10         0002 I 002470 B12A        0002 I 002471 B13
0002 I 002473 B16         0002 I 022545 B16A        0002 I 002474 B17B        0002 I 002462 B3          0002 I 002463 B5
0002 I 002464 B6          0002 I 002465 B8          0002 I 002466 B9          0000 I 025112 CASES       0000 I 024556 CDELTA
0002 I 015237 CFLAG       0000 I 025150 CGAMMA      0002 I 015750 CMAX        0000 I 000004 CODE        0000 I 025147 CTF
0002 I 002475 C1          0002 I 002557 C1A         0002 I 022546 C130        0002 I 022550 C5          0002 I 002641 C7A1
0002 I 002723 C7A2        0002 I 003005 C7B1        0002 I 003067 C7B2        0002 I 003315 C9          0000 I 025126 D
0002 I 017257 DELTA       0002 I 016735 DELTA       0000 I 025073 DIST        0000 I 000023 DLAT        0000 I 000045 DLON
0002 I 017256 D1          0002 I 003377 D1          0002 I 011267 D11         0002 I 012253 D12         0002 I 013242 D13
0002 I 022547 D130        0002 I 004363 D3          0002 I 005347 D4          0002 I 001324 D4FRST      0002 I 006333 D6
0000 I 025136 D6FRST      0000 I 025121 D6TOT       0002 I 007317 D8          0002 I 010303 D9          0000 R 025137 ENRTE
0000 R 000000 F           0000 R 000001 FC          0000 R 000002 FF          0000 I 025140 FRST        0000 I 025127 FS
0002 I 022635 GAMMA       0002 I 002310 I           0002 I 017247 IF          0002 I 022527 IHDR        0000 I 025107 II
0000 I 025116 IL          0002 I 022363 INEED       0002 I 017253 INREX       0002 I 017252 IPREC       0002 I 017245 IQT
0002 I 017246 IS          0002 I 001000 ISTAT       0000 I 025066 IS          0002 I 002311 J           0000 I 025110 JJ
0000 I 025122 K           0000 I 025113 KASES       0002 I 022357 KF          0000 I 025123 KK          0000 I 025117 KOPFAC
0000 I 025141 LAST        0000 I 025074 LAT         0000 I 025142 LEAVE       0002 I 002312 LGF         0000 I 000155 LIST
0002 I 017250 LMT         0002 I 000000 LOAD        0000 I 025075 LONG        0002 I 017254 LP          0002 I 017255 LPN
0002 I 015751 LVTIME      0002 I 022362 M           0000 I 025145 MAXTOS      0000 I 000626 MEED        0002 I 022360 MFLAG
0000 I 025134 MILES       0002 I 022634 MINC1       0000 I 025130 MIND        0000 I 000034 MLAT        0000 I 000056 MLON
0002 I 022361 N           0002 I 015130 NEED        0000 I 025143 NITEHR      0002 I 022632 NN          0002 I 014226 NO
0002 I 017251 NPREC       0000 I 025111 NS          0000 I 025135 NSEV        0002 I 001002 OPFAC       0000 R 025072 PI
0000 R 000067 PROB        0000 R 000100 R           0006 R 000000 RAND        0002 R 003151 RC7A        0002 R 003233 RC7B
0000 R 000111 RLAT        0000 R 000122 RLON        0000 R 025131 RN          0000 I 025071 S           0002 I 014310 SAVEI
0002 I 014620 SAVEJ       0000 I 025067 SERO        0002 I 001160 SOA3        0000 I 025120 STAMAX      0002 I 022633 STANO
0000 I 025146 SUMTOS      0002 I 022640 S1          0002 I 022641 S2          0000 R 000003 T           0002 I 022636 TE
0002 I 015440 TOS         0002 I 022637 TSM         0002 I 022643 X           0000 I 025105 XDELTA      0000 I 000133 XDIST
0000 I 025101 XLMT        0000 I 025103 XLOW        0000 I 025077 XPT         0000 I 000320 XSTA        0000 I 025124 XX
0002 I 022644 Y           0000 I 025106 YDELTA      0000 I 000144 YDIST       0000 I 025102 YLMT        0000 I 025104 YLOW
0000 I 025100 YPT         0000 I 000463 YSTA        0000 I 025125 YY          0000 R 025144 Z           0000 R 025076 ZBETA
0002 R 022642 ZMILE
```

```
00101     1*          REAL F,FC,FF,T
00103     2*          IMPLICIT INTEGER (A,B,C,D,E,G,O,S,T,X,Y)
00104     3*          COMMON LOAD(512),ISTAT(2),OPFAC(110),SOA3(100),D4FRST(50,10),I,J
00105     4*          COMMON LGF,A1A(50),A1B(50),A3,A4A,A4B,B3,B5,B6,B8,B9,B10,B12A,B13
00106     5*          COMMON A3A,B16,B17B,C1(50),C1A(50),C7A1(50),C7A2(50),C7B1(50),
00106     6*         1 C7B2(50),RC7A(50),RC7B(50),C9(50),D1(50,10),D3(50,10),D4(50,10),
00106     7*         2 D6(50,10),D8(50,10),D9(50,10),D11(50,10),D12(50,10),CFLAG,B(2),
00106     8*         3 D13(50,10),NO(50),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),CMAX,
00106     9*         4 LVTIME(50,10),DELTA(200),IQT,IS,IF,LMT,NPREC,IPREC,INREX,LP,
00106    10*         5 LPN,DX,DATA(1600),KF,MFLAG,N,M,INEED(100),IHDR(14),B16A,C130,D130
00107    11*          COMMON C5(50),NN,STANO,MINC1,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A3OLD
00110    12*          DIMENSION CODE(15),DLAT(9),MLAT(9),DLON(9),MLON(9),PROB(9),R(9),
00110    13*         1 RLAT(9),RLON(9),XDIST(9),YDIST(9),LIST(99),XSTA(99),YSTA(99),
00110    14*         2 MEED(200),BDCASE(10000),CDELTA(200)
00111    15*          DATA CODE/'ØØØ
```

```
00120     21*          READ (5,49) C130
00123     22*          READ (0,49) D130
00126     23*       49 FORMAT (A5)
00127     24*       49 FORMAT (I5)
00130     25*          READ (5,101) J ST,LAT,LONG,ZBETA,XPT,YPT,XLMT,YLMT,XLOW,YLOW,
00130     26*        1 XDELTA,YDELTA
00146     27*      101 FORMAT (3I6,F7.3,8I6)
00147     28*          READ (5,102) (50A3(I),I=1,70)
00155     29*      102 FORMAT (20I4)
00156     30*          READ (5,102) (INEED(I),I=1,100)
00164     31*        1 READ (5,107,END=2) J
00167     32*      107 FORMAT (I2)
00170     33*          I=J
00171     34*          IF (J.GT.9) I=J-10
00173     35*          READ (0,103) DLAT(I),MLAT(I),DLON(I),MLON(I),PROB(I)
00202     36*      103 FORMAT (2X,2(5X,I3,I2),6X,F5.3)
00202     37*    C
00202     38*    C  CONVERT DISTRICT CENTROIDS TO RADIANS
00202     39*    C
00203     40*          RLAT(I)=(DLAT(I)+MLAT(I)/60.0)*.017453
00204     41*          RLON(I)=(DLON(I)+MLON(I)/60.0)*.017453
00204     42*    C
00204     43*    C  CALCULATE DISTANCE FROM EACH DISTRICT TO DISTRICT ORIGIN(MILES)
00204     44*    C
00205     45*          XDIST(I)=((LONG-(DLON(I)*60+MLON(I)))*ZBETA)/60
00206     46*          YDIST(I)=(DLAT(I)*60+MLAT(I))-LAT
00207     47*          WRITE (6,500) J,RLAT(I),DLAT(I),MLAT(I),RLON(I),DLON(I),MLON(I),
00207     48*        1      PROB(I),XDIST(I),YDIST(I)
00223     49*      500 FORMAT (1X,I2,2(F12.8,2I6),F7.3,2I6)
00224     50*          GO TO 1
00225     51*        2 II=I
00226     52*          JJ=J
00227     53*          NS=1
00230     54*        3 READ (5,104,END=4) LIST(NS),XSTA(NS),YSTA(NS)
00235     55*      104 FORMAT (6X,I5,4X,2I6)
00236     56*          NS=NS+1
00237     57*          GO TO 3
00240     58*        4 NS=NS-1
00241     59*          READ (5,106) B(1),B(2)
00245     60*      106 FORMAT (2(D12.1X))
00246     61*          CASES=0
00247     62*          KASES=0
00250     63*          BD=0
00251     64*          BX=0
00252     65*          IL=0
00253     66*          MFLAG=0
00254     67*          KOPFAC=1
00255     68*          STAMAX=0
00256     69*          IQT=0
00257     70*          CALL NTRAN(7,2,256,LOAD(1),ISTAT(1))
00260     71*          IS=2
00261     72*          IF=257
00262     73*          LMT=482
00263     74*          NPREC=0
00264     75*          INREX=0
00265     76*          CALL READ ($20 )
00266     77*          LP=IF+32
00267     78*          GO TO 5
```

```
00267   79*   C
00267   80*   C  READ A CASE FROM MAP TAPE
00267   81*   C
00270   82*     10  IL=IL+1
00271   83*         CALL NUCASE (B*00)
00272   84*         CASES=CASES+1
00273   85*         IF (MFLAG.EQ.0.AND.(C5(1).EQ.SERO.OR.C5(1).EQ.A)) GO TO 105
00275   86*         KASES=KASES+1
00275   87*         DSTOT=0
00277   88*         TSM=0
00300   89*         N=0
00301   90*         M=0
00301   91*   C
00301   92*   C  CONVERT MULTI-UNIT CONTROL NUMBERS
00301   93*   C
00302   94*         DO 7 K=1,15
00305   95*         IF (A3A.NE.CODE(K)) GO TO 7
00307   96*         KK=K*10000
00310   97*         IF (K.GE.10) KK=(K+1)*1000
00312   98*         A3=A3+KK
00313   99*         IF (K.GE.10) A3=A3OLD+KK
00315  100*         GO TO 10
00316  101*       7 CONTINUE
00320  102*         XX=999
00321  103*         YY=999
00321  104*   C
00321  105*   C  CALCULATE CASE LOCATION X,Y
00321  106*   C
00322  107*      10 DO 13 I=1,CMAX
00325  108*         IF (C5(I).EQ.SERO.OR.C5(I).EQ.A) GO TO 13
00327  109*         IF ((C7A1(I).EQ.0.AND.C7A2(I).EQ.0).OR.(C7A1(I).EQ.99.AND.
00327  110*        1     C7A2(I).EQ.99)) GO TO 113
00331  111*         IF ((C7B1(I).EQ.0.AND.C7B2(I).EQ.0).OR.(C7B1(I).EQ.999.AND.
00331  112*        1     C7B2(I).EQ.99)) GO TO 113
00333  113*         IF (CFLAG.EQ.1) GO TO 11
00335  114*         X=((LONG-(C7B1(I)*60+C7B2(I)))*COBETA)/60
00336  115*         Y=(C7A1(I)*60+C7A2(I))-LAT
00337  116*         IF (X.GT.XLOW.AND.X.LT.XLMT.AND.Y.GT.YLOW.AND.Y.LT.YLMT) GO TO 14
00341  117*         X=2
00342  118*         Y=2
00343  119*         IF (MFLAG.EQ.0) GO TO 14
00345  120*         XX=X
00346  121*         YY=Y
00347  122*         GO TO 13
00350  123*      11 RC7A(I)=(C7A1(I)+C7A2(I)/60.0)*.017453
00351  124*         RC7B(I)=(C7B1(I)+C7B2(I)/60.0)*.017453
00352  125*         DO 111 D=1,II
00355  126*         IF (D.EQ.4.OR. .EQ.6) GO TO 111
00357  127*         IF ((II.EQ.JJ.AND.D.EQ.2).OR.(II.NE.JJ.AND.D.EQ.5)) GO TO 111
00361  128*         T=RLON(D)-RC7B(I)*COS(RLAT(D))
00362  129*         F=RLAT(D)-RC7A(I)
00363  130*         FC=ATAN(SIN(T)/TAN(F))
00364  131*         FF=(PI/2.0)-FC
00365  132*         FS=SIN(F)/SIN(FF)
00366  133*         IF (FS.GT.1.0) FS=1.0
00370  134*         IF (FS.LT.-1.0) FS=-1.0
00372  135*         R(D)=ASIN(FS)
00373  136*     111 CONTINUE
```

```
00375   137*          MIND=1
00375   138*          DO 112 D=1,II
00401   139*          IF (D.EQ.4.OR. .EQ.6) GO TO 112
00403   140*          IF ((II.EQ.JJ.AND.D.EQ.2).OR.(II.NE.JJ.AND.D.EQ.5)) GO TO 112
00405   141*          IF (ABS(R(D)).LT.ABS(R(MIND))) MIND=D
00407   142*      112 CONTINUE
00411   143*          A1A(I)=MIND
00412   144*          IF (II.NE.JJ)  1A(I)=A1A(I)+10
00414   145*          IF (A1A(I).EQ.DIST) A1B(I)=0
00416   146*          X=((LONG-(C7B1(I)*60+C7B2(I)))*COBETA)/60
00417   147*          Y=(C7A1(I)*60+C7A2(I))-LAT
00420   148*          GO TO 14
00421   149*      113 IF (CFLAG.EQ.1) GO TO 118
00423   150*          DO 114 J=1,NS
00426   151*          IF (A1B(I).EQ.DIST(J)) GO TO 116
00430   152*      114 CONTINUE
00432   153*          WRITE (6,115) A1B(I)
00435   154*      115 FORMAT (1H ,'UNKNOWN LOCATION FOR OPFAC ',I6)
00436   155*          X=XPT
00437   156*          Y=YPT
00440   157*          GO TO 117
00441   158*      116 X=XSTA(J)+XDELTA
00442   159*          Y=YSTA(J)+YDELTA
00443   160*      117 IF (MFLAG.EQ.0) GO TO 14
00445   161*          GO TO 13
00446   162*      118 RN=RAND(0,B)
00447   163*          DO 119 D=II,1,-1
00452   164*          IF (D.EQ.4.OR.D.EQ.6) GO TO 119
00454   165*          IF ((II.EQ.JJ.AND.D.EQ.2).OR.(II.NE.JJ.AND.D.EQ.5)) GO TO 119
00456   166*          IF (RN.LT.PROB(D)) ASSD=D
00460   167*      119 CONTINUE
00462   168*          A1A(I)=ASSD
00463   169*          IF (II.NE.JJ) A1A(I)=A1A(I)+10
00465   170*          IF (A1A(I).EQ.DIST) A1B(I)=0
00467   171*          X=XDIST(ASSD)
00470   172*          Y=YDIST(ASSD)
00471   173*          GO TO 14
00472   174*       13 CONTINUE
00474   175*          IF (XX.NE.999.AND.YY.NE.999) GO TO 122
00476   176*          DO 120 J=1,NS
00501   177*          IF (A1B(1).EQ. IST(J)) GO TO 121
00503   178*      120 CONTINUE
00505   179*          WRITE (6,115) A1B(1)
00510   180*          X=XPT
00511   181*          Y=YPT
00512   182*          GO TO 14
00513   183*      121 X=XSTA(J)+XDELTA
00514   184*          Y=YSTA(J)+YDELTA
00515   185*          GO TO 14
00516   186*      122 X=XX
00517   187*          Y=YY
00517   188*    C
00517   189*    C  SET DATE AND TIME OF NOTIFICATION 'MINC1'
00517   190*    C
00520   191*       14 MINC1=C1(1)
00520   192*    C
00520   193*    C  CONVERT 'A1B(I)' TO 'STAND', THE PRIMARY STATION
00520   194*    C
```

```
00621   195*          IF (WFLAG.EQ.0.AND.CFLAG.EQ.0)  GO TO 20
00623   196*          IF (A1B(1).EQ.KOPFAC) GO TO 21
00625   197*          DO 15 I=1,STAMAX
00630   198*          IF (A1B(1).EQ.OPFAC(I)) GO TO 17
00632   199*       1  CONTINUE
00634   200*          GO TO 20
00635   201*      17  STANO=I
00636   202*          IF (KOPFAC.EQ.0.AND.OPFAC(STAMAX-1).EQ.0) STANO=STANO-1
00640   203*          IF (A1B(1).EQ.0) STANO=0
00642   204*          KOPFAC=A1B(1)
00643   205*          GO TO 21
00644   206*      20  IF (A1B(1).EQ.KOPFAC) GO TO 21
00646   207*          STAMAX=STAMAX+1
00647   208*          OPFAC(STAMAX)=A1B(1)
00650   209*          STANO=STAMAX
00651   210*          IF (KOPFAC.EQ.0) STANO=STANO-1
00653   211*          IF (A1B(1).EQ.0) STANO=0
00655   212*          KOPFAC=A1B(1)
00656   213*      21  DO 25 I=1,CMAX
00661   214*          IF (C5(I).EQ.5.AND.C5(I).EQ.4) GO TO 25
00663   215*          IF (NO(I).EQ.0) GO TO 25
00663   216*   C
00663   217*   C  CALCULATE SEARCH  ILES 'TSM'
00663   218*   C
00665   219*          K=NO(I)
00666   220*          DO 24 J=1,K
00671   221*          IF (D1(I,J).GE.70) GO TO 24
00673   222*          IF (D6(I,J).EQ.0) GO TO 24
00675   223*          D6TOT=D6TOT+D6(I,J)
00676   224*          BOAT=D1(I,J)
00677   225*          TSM=TSM+(D6(I,J)*SDA3(BOAT))
00600   226*      24  CONTINUE
00602   227*      25  CONTINUE
00604   228*          TSM=(TSM+9)/10
00604   229*   C
00604   230*   C  CLACULATE MILES OFF-SHORE, 'MILES'
00604   231*   C
00605   232*          ZMILE=B16
00606   233*          IF (B16A.EQ.5) ZMILE=0
00610   234*          IF (B16.GE.1.AND.B16.LE.6) ZMILE=.35
00612   235*          IF (B16.EQ.7.OR.B16.EQ.8) ZMILE=3
00614   236*          IF (B16.EQ.9) ZMILE=8
00616   237*          IF (B16.EQ.999) ZMILE=.95
00620   238*          MILES=ZMILE*10
00620   239*   C
00620   240*   C  CONVERT SEVERITY 'NSEV'
00620   241*   C
00621   242*          NSEV=B17B
00622   243*          IF (B17B.EQ.3) NSEV=4
00624   244*          IF (B17B.EQ.4) NSEV=3
00626   245*          IF (NSEV.LT.1) NSEV=1
00630   246*          IF (NSEV.GT.5) NSEV=5
00632   247*          IF (D6TOT.GT.5) GO TO 50
00632   248*   C
00632   249*   C  SHORT SEARCH CASE
00632   250*   C
00634   251*          S1=0
00635   252*          S2=1
```

```
00636   255*          D6FRST=0
00637   256*          DO 27 I=1,CMAX
00642   257*          IF (C5(I).EQ.5 .OR.C5(I).EQ.4) GO TO 27
00644   258*          IF (ND(I).EQ.0) GO TO 27
00646   259*          K=ND(I)
00647   260*          IF (D6FRST.EQ.1) GO TO 261
00651   261*          DO 260 J=1,K
00654   262*          IF (D1(I,J).GE.70) GO TO 260
00656   263*          IF (D6(I,J).GT.0) GO TO 26
00660   264*          D6FRST=1
00661   265*          GO TO 261
00662   266*       26 CONTINUE
00664   267*          GO TO 27
00665   268*      261 DO 26 J=1,K
00670   269*          IF (D1(I,J).GE.70) GO TO 26
00672   270*          IF (D6(I,J).GT.0) S2=2
00674   271*       26 CONTINUE
00676   272*       27 CONTINUE
00700   273*          IF (D6TOT.EQ.0) S2=0
00702   274*       35 DO 351 I=1,CMAX
00705   275*          IF (C5(I).EQ.5.OR.C5(I).EQ.4) GO TO 351
00707   276*          IF (ND(I).EQ.0) GO TO 351
00711   277*          K=ND(I)
00712   278*          DO 35 J=1,K
00715   279*          IF (D1(I,J).GE.70) GO TO 35
00717   280*          D4FRST(I,J)=D4(I,J)+D6(I,J)
00720   281*          N=N+1
00720   282*    C
00720   283*    C  DETERMINE 'NEED(N)', ','MM','TOS(N)'
00720   284*    C
00721   285*          CALL NEEDS
00722   286*          IF (NEED(N).EQ.0.OR.NEED(N).GT.19) NEED(N)=18
00724   287*          IF (NEED(N).EQ.10.OR.NEED(N).EQ.15.OR.NEED(N).EQ.17.OR.NEED(N)
00724   288*         1     .EQ.19) GO TO 29
00726   289*          SAVE1(N)=I
00727   290*          SAVEJ(N)=J
00730   291*          ENRTE=D4(I,J)-D3(I,J)
00731   292*          IF (ENRTE.LT.0) ENRTE=0
00733   293*          TOS(N)=D8(I,J)-D6(I,J)-ENRTE*2*D9(I,J)
00734   294*          IF (TOS(N).LT. ) TOS(N)=0
00736   295*          LVTIME(I,J)=D4(I,J)+D6(I,J)+TOS(N)
00737   296*          GO TO 35
00740   297*       29 N=N-1
00741   298*       30 LVTIME(I,J)=D4FRST(I,J)
00742   299*          M=M+1
00743   300*          MEED(M)=NEED(N+1)
00744   301*       35 CONTINUE
00746   302*      351 CONTINUE
00750   303*          GO TO 70
00750   304*    C
00750   305*    C  LONG SEARCH - CALCULATE 'TE' AND 'S1'
00750   306*    C
00751   307*       50 IF (CMAX.GT.1.OR.ND(1).GT.1) GO TO 511
00753   308*          TE=D6TOT
00754   309*      511 IF (D6TOT.GT.140) GO TO 55
00756   310*          S1=1
00757   311*          GO TO 60
00760   312*      511 FRST=0
```

```
00751   311*          LAST=0
00752   312*          DO 512 I=1,CMAX
00755   313*          IF (C5(I).EQ.SERO.OR.C5(I).EQ.0) GO TO 512
00757   314*          IF (NO(I).EQ.0) GO TO 512
00771   315*          K=NO(I)
00772   316*          DO 51 J=1,K
00775   317*          IF (D1(I,J).LT.70) GO TO 52
00777   318*       51 CONTINUE
01001   319*      512 CONTINUE
01003   320*          GO TO 520
01004   321*       52 FRST=D4(I,J)
01005   322*          LAST=FRST+D6(I,J)
01006   323*      52  DO 54 I=1,CMAX
01011   324*          IF (C5(I).EQ.SERO.OR.C5(I).EQ.0) GO TO 54
01013   325*          IF (NO(I).EQ.0) GO TO 54
01015   326*          K=NO(I)
01016   327*          DO 53 J=1,K
01021   328*          IF (D1(I,J).GE.70) GO TO 53
01023   329*          IF (D4(I,J).LT.FRST) FRST=D4(I,J)
01025   330*          LEAVE=D4(I,J)+D6(I,J)
01026   331*          IF (LEAVE.GT.LAST) LAST=LEAVE
01030   332*       5  CONTINUE
01032   333*       54 CONTINUE
01034   334*          TE=LAST-FRST
01035   335*          IF (D6TOT.GT.140) GO TO 55
01037   336*          NITEHR=0
01040   337*          GO TO 56
01041   338*       55 IF (TE.LE.240) NITEHR=TE-140
01043   339*          IF (TE.GT.240) NITEHR=TE/2.4
01045   340*       56 Z=D6TOT
01046   341*          S1=Z/(TE-NITEH )+.99
01047   342*          IF (S1.GT.10) S1=10
01051   343*       61 S2=0
01052   344*          GO TO 28
01052   345*    C
01052   346*    C  CALCULATE 'DELTA(N)' AND 'GAMMA'
01052   347*    C
01053   348*       70 MAXTOS=0
01054   349*          SUMTOS=0
01055   350*          TE=0
01056   351*          CTE=0
01057   352*          IF (N.EQ.0) GO TO 76
01061   353*          I=SAVEI(1)
01062   354*          J=SAVEJ(1)
01063   355*          FRST=D4FRST(I,J)
01064   356*          LAST=LVTIME(I,J)
01065   357*          IF (N.EQ.1) GO TO 74
01067   358*          DO 71 K=2,N
01072   359*          I=SAVEI(K)
01073   360*          J=SAVEJ(K)
01074   361*          IF (D4FRST(I,J).LT.FRST) FRST=D4FRST(I,J)
01076   362*          IF (LVTIME(I,J).GT.LAST) LAST=LVTIME(I,J)
01100   363*       71 CONTINUE
01102   364*       74 TE=LAST-FRST
01103   365*          CTE=TE
01104   366*          IF (TE.LT.0) TE=0
01106   367*          DO 75 K=1,N
01111   368*          I=SAVEI(K)
```

```
01112   369*          JE=AVLJ(K)
01113   370*          IF (TE.NE.TOS( )) GO TO 72
01115   371*          DELTA(K)=0
01116   372*          CDELTA(K)=0
01117   373*          GO TO 73
01120   374*       72 DELTA(K)=(100*(ERST(I,J)-ERST))/(TE-TOS(K))
01121   375*          CDELTA(K)=DELTA(K)
01122   376*          IF (DELTA(K).GT.99) DELTA(K)=99
01124   377*          IF (DELTA(K).LT.0) DELTA(K)=0
01126   378*       73 IF (TOS(K).GT.MAXTOS) MAXTOS=TOS(K)
01130   379*          SUMTOS=SUMTOS+TOS(K)
01131   380*       74 CONTINUE
01133   381*       75 NN=1
01134   382*          IF (SUMTOS.NE.MAXTOS) GO TO 78
01136   383*          GAMMA=0
01137   384*          CGAMMA=0
01140   385*          GO TO 80
01141   386*       78 GAMMA=(100*(TE-MAXTOS))/(SUMTOS-MAXTOS)
01142   387*          CGAMMA=GAMMA
01143   388*          IF (GAMMA.GT.99) GAMMA=99
01145   389*          IF (GAMMA.LT.0) GAMMA=0
01147   390*       80 IF (M.EQ.0) GO TO 90
01151   391*          NN=N+1
01152   392*          TOS(NN)=0
01153   393*          DELTA(NN)=0
01154   394*          NEED(NN)=NEED(1)
01155   395*          IF (M.EQ.1) GO TO 90
01157   396*          DO 81 I=2,M
01158   397*          IF (NEED(I).EQ.10) GO TO 81
01164   398*          IF (NEED(NN).EQ.10.OR.NEED(NN).EQ.19.OR.NEED(I).EQ.17)
01164   399*         1     NEED(NN)=NEED(I)
01166   400*       81 CONTINUE
01170   401*       90 A4=A4A*10+A4B
01171   402*          IF (B13.EQ.50) B13=66
01173   403*          IF (B13.EQ.51) B13=101
01175   404*          IF (B13.EQ.52) B13=201
01177   405*          IF (B13.GE.60.AND.B13.LE.99) B13=0
01201   406*          IF (NN.NE.0) GO TO 97
01203   407*          IF (S1.GT.0.OR.S2.GT.0) GO TO 95
01205   408*          KASES=KASES-1
01206   409*          BD=BD+1
01207   410*          IF (BD.GT.100) GO TO 5
01211   411*          DO 91 D=1,DX
01214   412*          BX=BX+1
01215   413*          BDCASE(BX)=DATA(D)
01216   414*       91 CONTINUE
01220   415*          GO TO 5
01221   416*       95 NEED(1)=0
01222   417*          TOS(1)=0
01223   418*          DELTA(1)=0
01224   419*       97 IF (STAND.LE.7.OR.A1B(1).EQ.D130)
01224   420*         1WRITE (6,95)  A1A(1),STAND,A3,AB,MINC1,B3,B5,B6,B8,B9,B10,B12A,
01224   421*         2 B13,NSEV
01245   422*       95 FORMAT (1H0,6I6,I12,7I6)
01246   423*          IF (STAND.LE.7.OR.A1B(1).EQ.D130)
01246   424*         1WRITE (6,96) NN,N,M,GAMMA,TE,S1,S2,TSM,MILES,X,Y,CGAMMA,CTF
01266   425*       96 FORMAT (1H ,13I6)
01267   426*          IF (STAND.GT.7.AND.A1B(1).NE.D130) GO TO 99
```

```
01271   427*          DO 94 K=1,NN
01274   428*          WRITE (6,93) NEED(K),TOS(K),DELTA(K),COPLTS(K)
01302   429*       93 FORMAT (1H ,4I6)
01303   430*          IF (K.GT.N) GO TO 94
01305   431*          I=SAVEI(K)
01306   432*          J=SAVEJ(K)
01307   433*          WRITE (6,93) DR(I,J),DR(I,J)
01313   434*       93 FORMAT (1H+,T26,2I6)
01314   435*       94 CONTINUE
01315   436*       95 WRITE (8) A1A(J),STANO,A3,A4,ATC1,B3,B5,B6,B8,B10,B12,B13,
01316   437*         1 NSEV,NN,N,M,GAMMA,TE,S1,S2,TSM,MILES,X,Y,NEED(1),TOS(1),DELTA(1)
01354   438*          IF (NN.LE.1) GO TO 5
01356   439*          WRITE (8) ((NEED(K),TOS(K),DELTA(K)),K=2,NN)
01366   440*          GO TO 5
01367   441*      200 WRITE (6,201)
01371   442*      201 FORMAT (1H0,'END')
01372   443*          WRITE (6,203) CASES,KASES,BD,IL
01400   444*      203 FORMAT (1H0,'CASES:',6X,'TOTAL=',I6,6X,'GOOD=',I6,6X,'BAD=',I6,6X,
01400   445*         1 'ELIMINATED=',I6/)
01401   446*          WRITE (6,205) (BDCASE(B),B=1,8X)
01407   447*      205 FORMAT (2(16(1X,A6)/))
01410   448*          DO 202 I=1,28
01413   449*      202 LOAD(I)=99
01415   450*          WRITE (8) (LOAD(I),I=1,28)
01423   451*          WRITE (6,300)
01425   452*      300 FORMAT (1H1,'OPFAC',4X,'STATION'/)
01426   453*          DO 301 I=1,STAMAX
01431   454*      301 WRITE (6,302) OPFAC(I),I
01436   455*      302 FORMAT (1H ,I5,6X,I3)
01437   456*          STOP
01440   457*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   1 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   1 SEC.


TOTAL COMPILATION TIME =   5  SEC
```

```
@IT F,R READ,READ                                                                    02 JUN 71          15:22:01.656
UNIVAC 1108 FORTRAN V LEVEL   2206  01  F50143
THIS COMPILATION WAS DONE ON 06 JUN 71 AT 15:22:41


   SUBROUTINE READ        ENTRY POINT 000232

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000242
         0000   *DATA   000036
         0002   *BLANK  022646


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NTRAN
         0004   NWD1$
         0005   NIO2$
         0006   NSTOP$
         0007   NERR4$
         0010   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000006 1DOF        0001   000161 12L         0001   000127 1546        0001   000016 2L          0001   000025 3L
    0001   000053 4L          0001   000106 5L          0001   000112 6L          0002 I 002313 A1A         0002 I 002375 A1B
    0002 I 002457 A3          0002 I 002472 A3A         0002 I 022645 A3OLD       0002 I 002460 A4A         0002 I 002461 A4B
    0002 I 013240 B           0000 I 000000 BLANK       0002 I 002467 B10         0002 I 002470 B12A        0002 I 002471 B13
    0002 I 002473 B16         0002 I 022545 B16A        0002 I 002474 B17B        0002 I 002462 B3          0002 I 002463 B5
    0002 I 002464 B6          0002 I 002465 B8          0002 I 002466 B9          0002 I 013237 CFLAG       0002 I 015750 CMAX
    0002 I 002475 C1          0002 I 002557 C1A         0002 I 022546 C130        0002 I 022550 C5          0002 I 002641 C7A1
    0002 I 002723 C7A2        0002 I 003005 C7B1        0002 I 003067 C7B2        0002 I 003315 C9          0002 I 017257 DATA
    0002 I 016735 DELTA       0000 I 000003 DERR        0002 I 017256 DX          0002 I 003377 D1          0002 I 011267 D11
    0002 I 012253 D12         0002 I 013242 D13         0002 I 022547 D130        0002 I 004363 D3          0002 I 005347 D4
    0002 I 001524 D4FRST      0002 I 006333 D6          0002 I 007317 D8          0002 I 010303 D9          0002 I 022635 GAMMA
    0002 I 002310 I           0002 I 017247 IF          0002 I 022527 IHDR        0002 I 022363 INFED       0002 I 017253 INREX
    0002 I 017252 IPREC       0002 I 017245 IQT         0002 I 017246 IS          0002 I 001000 ISTAT       0000 I 000001 IT
    0002 I 002311 J           0000 I 000002 JF          0002 I 022357 KF          0002 I 002312 LGF         0002 I 017250 LMT
    0002 I 000000 LOAD        0002 I 017254 LP          0002 I 017255 LPN         0002 I 015751 LVTIME      0000 I 000005 LX
    0002 I 022362 M           0002 I 022360 MFLAG       0002 I 022634 MINC1       0002 I 022361 N           0002 I 015130 NEED
    0000 I 000004 NEXT        0002 I 022632 NN          0002 I 014226 NO          0002 I 017251 NPREC       0002 I 001002 OPFAC
    0002 R 003151 RC7A        0002 R 003233 RC7B        0002 I 014310 SAVEI       0002 I 014620 SAVEJ       0002 I 001160 SOA3
    0002 I 022633 STANO       0002 I 022640 S1          0002 I 022641 S2          0002 I 022636 TE          0002 I 015440 TOS
    0002 I 022637 TSM         0002 I 022643 X           0002 I 022644 Y           0002 R 022642 ZMILE


00101     1*          SUBROUTINE READ  (S)
00103     2*          IMPLICIT INTEGER (A,B,C,D,F,G,O,S,T,X,Y)
00104     3*          COMMON LOAD(512),ISTAT(2),OPFAC(110),SOA3(100),D4FRST(50,10),I,J
00105     4*          COMMON LGF,A1A(50),A1B(50),A3,A4A,A4B,B3,B5,B6,B8,B9,B10,B12A,B13
00106     5*          COMMON A3A,B16,B17B,C1(50),C1A(50),C7A1(50),C7A2(50),C7B1(50),
00108     6*         1 C7B2(50),RC7A(50),RC7B(50),C9(50),D1(50,10),D3(50,10),D4(50,10),
```

```
00106     7*          2 D6(50,10),D8(50,10),D9(50,10),D11(50,10),D12(50,10),CFLAG,B(2),
00106     8*          3 D13(50,10),ND(50),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),CMAX,
00106     9*          4 LVTIMC(50,10),DELTA(200),IQT,IS,IF,LMT,NPREC,IPREC,INREX,LP,
00106    10*          5 LPN,DX,DATA(1600),KF,MFLAG,N,M,INEED(100),IHDR(14),B16A,C130,D130
00107    11*            COMMON C5(50),NN,STAND,MINC1,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A30LD
00110    12*            DATA BLANK/'      '/
00112    13*          1 IT=3-IS
00113    14*            JF=258-IF
00114    15*            KF=IF+IF-2
00115    16*          2 IF (ISTAT(IT)+1),2,6
00120    17*            IF (ISTAT(IT)+2),5,6
00123    18*            DERR=1
00124    19*          3 CALL NTRAN (7,22)
00125    20*            CALL NTRAN (7,7,-1)
00126    21*            CALL NTRAN (7,2,256,LOAD(JF),ISTAT(IT))
00127    22*          4 IF (ISTAT(IT)+1),4,6
00132    23*            IF (ISTAT(IT)+2),5,6
00135    24*            DERR=DERR+1
00136    25*            IF (DERR.LT.4) GO TO 3
00140    26*            NEXT=INREX+1
00141    27*            WRITE (6,100) NEXT
00144    28*        100 FORMAT (1H ,'UNABLE TO READ RECORD ',I6)
00145    29*            STOP
00146    30*          5 RETURN 1
00147    31*          6 IF (IQT.EQ.1) RETURN
00151    32*         10 IF (NPREC.GT.0) GO TO 12
00153    33*            DO 11 I=1,14
00156    34*            LX=JF+I-1
00157    35*         11 IHDR(I)=LOAD(LX)
00161    36*            IPREC=IHDR(10)
00162    37*            MFLAG=0
00163    38*            CFLAG=0
00164    39*            IF (IHDR(2).EQ.BLANK) MFLAG=1
00166    40*            IF (IHDR(2).EQ.C130) CFLAG=1
00170    41*         12 NPREC=NPREC+1
00171    42*         13 INREX=INREX+1
00172    43*            CALL NTRAN (7,2,256,LOAD(IF),ISTAT(IS))
00173    44*            IS=IT
00174    45*            IF=JF
00175    46*            LMT=708-LMT
00176    47*            RETURN
00177    48*            END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   1 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   1  SEC
```

```
WIT FOR NEEDS,NEEDS                                                                    09 JUN 71          15:22:42.797
UNIVAC 1108 FORTRAN V LEVEL   2206  018 F50139
THIS COMPILATION WAS DONE 01 05 JUN 71 AT 15:22:42


   SUBROUTINE NEEDS      ENTRY POINT 000366

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000374
         0000    *DATA    000014
         0002    *BLANK   022646


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000236 10L        0001    000312 11L        0001    000250 12L        0001    000346 13L        0001    000351 14L
    0001    000122 2L         0001    000127 5L         0001    000224 7L         0001    000231 8L         0002 I 002313 A1A
    0002 I 002375 A1B         0002 I 002457 A3          0002 I 002472 A3A         0002 I 022645 A3OLD       0002 I 002460 A4A
    0002 I 002461 A4B         0002 I 013240 B           0002 I 002467 B10         0002 I 002470 B12A        0002 I 002471 B13
    0002 I 002473 B16         0002 I 022545 B16A        0002 I 002474 B17B        0002 I 002462 B3          0002 I 002463 B5
    0002 I 002464 B6          0002 I 002465 B8          0002 I 002466 B9          0002 I 013237 CFLAG       0002 I 015750 CMAX
    0002 I 002475 C1          0002 I 002557 C1A         0002 I 022546 C130        0002 I 022550 C5          0002 I 002641 C7A1
    0002 I 002723 C7A2        0002 I 003005 C7B1        0002 I 003067 C7B2        0002 I 003315 C9          0000 I 000000 D
    0002 I 017257 DATA        0002 I 016735 DELTA       0002 I 017256 DX          0002 I 003377 D1          0002 I 011267 D11
    0002 I 012253 D12         0002 I 013242 D13         0002 I 022547 D130        0002 I 004363 D3          0002 I 005347 D4
    0002 I 001324 D4FRST      0002 I 006333 D6          0002 I 007317 D8          0002 I 010303 D9          0002 I 022635 GAMMA
    0002 I 002510 I           0002 I 017247 IF          0002 I 022527 IHDR        0002 I 022363 INFED       0002 I 017253 INREX
    0002 I 017252 IPREC       0002 I 017245 IQT         0002 I 017246 IS          0002 I 001000 ISTAT       0002 I 002311 J
    0002 I 022357 KF          0002 I 002312 LGF         0002 I 017250 LMT         0002 I 000000 LOAD        0002 I 017254 LP
    0002 I 017255 LPN         0002 I 015751 LVTIME      0002 I 022362 M           0002 I 022360 MFLAG       0002 I 022634 MINC1
    0002 I 022361 N           0002 I 015130 NEED        0002 I 022632 NN          0002 I 014226 NO          0002 I 017251 NPREC
    0002 I 001002 OPFAC       0002 R 003151 RC7A        0002 R 003233 RC7B        0002 I 014310 SAVEI       0002 I 014620 SAVEJ
    0002 I 001160 S0A3        0002 I 022633 STANO       0002 I 022640 S1          0002 I 022641 S2          0002 I 022636 TE
    0002 I 015440 TOS         0002 I 022637 TSM         0002 I 022643 X           0002 I 022644 Y           0002 R 022642 ZMILE


00101      1*          SUBROUTINE NEEDS
00103      2*          IMPLICIT INTEGER (A,B,C,D,E,G,O,S,T,X,Y)
00104      3*          COMMON LOAD(512),ISTAT(2),OPFAC(110),S0A3(100),D4FRST(50,10),I,J
00105      4*          COMMON LGF,A1A(50),A1B(50),A3,A4A,A4B,B3,B5,B6,B8,B9,B10,B12A,B13
00106      5*          COMMON A3A,B16,B17B,C1(50),C1A(50),C7A1(50),C7A2(50),C7B1(50),
00106      6*         1 C7B2(50),RC7A(50),RC7B(50),C9(50),D1(50,10),D3(50,10),D4(50,10),
00106      7*         2 D6(50,10),D8(50,10),D9(50,10),D11(50,10),D12(50,10),CFLAG,B(2),
00106      8*         3 D13(50,10),NO(50),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),CMAX,
00106      9*         4 LVTIME(50,10),DELTA(200),IQT,IS,IF,LMT,NPREC,IPREC,INREX,LP,
00106     10*         5 LPN,DX,DATA(1600),KF,MFLAG,N,M,INFED(100),IHDR(14),B16A,C130,D130
00107     11*          COMMON C5(50), D,STANO,MINC1,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A3OLD
00110     12*          DEDI(I,J)/10
```

```
00111    13*          IF (D.EQ.0) GO TO 10
00113    14*          IF (D11(I,J).E .0) GO TO 5
00115    15*          IF (D12(I,J).E .63.OR.D12(I,J).EQ.65.OR.D12(I,J).EQ.6[illegible]) GO TO 2
00117    16*          IF (D12(I,J).E .64.OR.D13(I,J).EQ.63.OR.D13(I,J).EQ.65) GO TO 2
00121    17*          IF (D13(I,J).EQ.66.OR.D13(I,J).EQ.69) GO TO 2
00123    18*          IF (D12(I,J).EQ.76.OR.D13(I,J).EQ.76) GO TO 2
00125    19*          D=D11(I,J)
00126    20*          NEED(N)=INEED(D)
00127    21*          RETURN
00130    22*        2 NEED(N)=19
00131    23*          RETURN
00132    24*        5 IF (D12(I,J).E .0) GO TO 8
00134    25*          IF (D13(I,J).EQ.63.OR.D13(I,J).EQ.65) GO TO 7
00136    26*          IF (D13(I,J).EQ.66.OR.D13(I,J).EQ.69.OR.D13(I,J).EQ.76) GO TO 7
00140    27*          D=D12(I,J)
00141    28*          NEED(N)=INEED(D)
00142    29*          IF (D12(I,J).EQ.68.AND.B13.GE.60.AND.B13.LT.70) NEED(N)=11
00144    30*          RETURN
00145    31*        7 NEED(N)=15
00146    32*          RETURN
00147    33*          NEED(N)=14
00150    34*          RETURN
00151    35*       10 IF (D11(I,J).E .0) GO TO 12
00153    36*          D=D11(I,J)
00154    37*          NEED(N)=INEED(D)
00155    38*          RETURN
00156    39*       12 IF (D12(I,J).EQ.0) GO TO 14
00160    40*          IF (D12(I,J).EQ.63.OR.D12(I,J).EQ.65) GO TO 13
00162    41*          IF (D12(I,J).EQ.66.OR.D12(I,J).EQ.69.OR.D12(I,J).EQ.76) GO TO 13
00164    42*          D=D12(I,J)
00165    43*       11 NEED(N)=INEED(D)
00166    44*          IF (B13.GE.60.AND.B13.LT.70.AND.D.EQ.68) NEED(N)=17
00170    45*          RETURN
00171    46*       13 D=68
00172    47*          GO TO 11
00173    48*       14 NEED(N)=18
00174    49*          RETURN
00175    50*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   1 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   2  SEC
```

```
SUBROUTINE NUCASE     ENTRY POINT 000342

STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE   000352
      0000   *DATA   000031
      0002   *BLANK  022646


EXTERNAL REFERENCES (BLOCK, NAME)

      0003   READ
      0004   FIELD
      0005   FIELD3
      0006   FIELD2
      0007   NERR4$
      0010   NERR3$


STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001   000054 127G       0001   000315 199L       0001   000322 200L       0001   000000 201L       0001   000032 205L
 0001   000114 213L       0001   000154 214L       0001   000171 215L       0001   000175 216L       0001   000222 217L
 0001   000237 220L       0001   000267 221L       0001   000272 240L       0001   000133 250L       0002 I 002313 A1A
 0002 I 002375 A1B        0002 I 002457 A3         0002 I 002472 A3A        0002 I 022645 A30LD      0002 I 002460 A4A
 0002 I 002461 A4B        0002 I 013240 B          0002 I 002467 B10        0002 I 002470 B12A       0002 I 002471 B13
 0002 I 002473 B16        0002 I 022545 B16A       0002 I 002474 B17B       0002 I 002462 B3         0002 I 002463 B5
 0002 I 002464 B6         0002 I 002465 B8         0002 I 002466 B9         0002 I 013237 CFLAG      0002 I 015750 CMAX
 0002 I 002475 C1         0002 I 002557 C1A        0002 I 022546 C130       0002 I 022550 C5         0002 I 002641 C7A1
 0002 I 002723 C7A2       0002 I 003005 C7B1       0002 I 003067 C7B2       0002 I 003315 C9         0002 I 017257 DATA
 0002 I 016735 DELTA      0002 I 017256 DX         0000 I 000005 DZ         0002 I 003377 D1         0002 I 011267 D11
 0002 I 012253 D12        0002 I 013242 D13        0002 I 022547 D130       0002 I 004363 D3         0002 I 005347 D4
 0002 I 001524 D4FRST     0002 I 006333 D6         0002 I 007317 D8         0002 I 010303 D9         0002 I 022635 GAMMA
 0002 I 002510 I          0000 I 000003 IEND       0002 I 017247 IF         0002 I 022527 IHDR       0002 I 022363 INEED
 0002 I 017253 INREX      0000 I 000001 IPARTS     0002 I 017252 IPREC      0002 I 017245 IQT        0002 I 017246 IS
 0002 I 001000 ISTAT      0000 I 000007 IX         0000 I 000013 IZ         0000 I 000011 I1         0000 I 000012 I2
 0002 I 002511 J          0000 I 000004 K          0000 I 000010 KD3        0002 I 022357 KF         0000 I 000002 LAST
 0002 I 002512 LGF        0002 I 017250 LMT        0002 I 000000 LOAD       0002 I 017254 LP         0002 I 017255 LPN
 0002 I 015751 LVTIME     0002 I 022362 M          0002 I 022360 MFLAG      0002 I 022634 MINC1      0002 I 022361 N
 0000 I 000006 NDX        0002 I 015130 NEED       0000 I 000000 NLGF       0002 I 022632 NN         0002 I 014226 NO
 0002 I 017251 NPREC      0002 I 001002 OPFAC      0002 R 003151 RC7A       0002 R 003233 RC7B       0002 I 014310 SAVEI
 0002 I 014620 SAVEJ      0002 I 001160 SOA3       0002 I 022633 STANO      0002 I 022640 S1         0002 I 022641 S2
 0002 I 022636 TE         0002 I 015440 TOS        0002 I 022637 TSM        0002 I 022643 X          0002 I 022644 Y
 0002 R 022642 ZMILE
```

```
00101     1*          SUBROUTINE NUCASE ($)
00103     2*          IMPLICIT INTEGER (A,B,C,D,E,G,O,S,T,X,Y)
00104     3*          COMMON LOAD(512),ISTAT(2),OPFAC(110),SOA3(100),D4FRST(50,10),T,J
```

```
00105    4*        COMMON LGF,A1A(50),B13(50),A3,A4A,A4B,B3,B5,B6,B8,B9,B10,B12A,B13
00106    5*        COMMON A3A,B16,B17B,C1(50),C1A(50),C7A1(50),C7A2(50),C7B1(50),
00108    6*       1 C7B2(50),C7A(50),C7B(50),C3(50),D1(50,10),D3(50,10),D4(50,10),
00108    7*       2 D5(50,10),D6(50,10),D8(50,10),D11(50,10),D12(50,10),CFLAG,B(2),
00106    8*       3 D13(50,10),ND(50),SAVEI(200),SAVEJ(200),NFED(200),TOS(200),CMAX,
00106    9*       4 LVTIME(50,10),DELTA(250),IST,IF,IF,LMT,NPREC,IPREC,IPREV,LP,
00106   10*       5 LPN,DX,DATA(1200),KF,MFLAG,N,M,INFED(100),IHOR(10),B16A,C130,D130
00107   11*        COMMON C5(50),DN,STAND,MINC1,GAMMA,TE,TSM,S1,S2,ZMILE,X,Y,A30LD
00110   12*    201 IF (LOAD(LP).E .0) GO TO 220
00112   13*        NLGF=0
00113   14*        DX=0
00114   15*        LGF=FLD(2,4,LOAD(LP))
00115   16*        IF (MFLAG.EQ.1) LGF=LOAD(LP)
00117   17*        IF (MFLAG.EQ.1) IPARTS=LOAD(LP+1)
00121   18*    203 LPN=LP+32
00122   19*        LAST=LPN-1
00123   20*        IEND=IF+255
00124   21*        IF (LAST.GT.IEND) LAST=IEND
00126   22*        DO 210 K=LP,LAST
00131   23*        DX=DX+1
00132   24*    210 DATA(DX)=LOAD(K)
00134   25*        NLGF=NLGF+1
00135   26*        IF (LPN.LT.LMT) GO TO 213
00137   27*        IF (NPREC.EQ.IPREC) NPREC=0
00141   28*        CALL READ($200)
00142   29*        LPN=LPN-KF
00143   30*        IF (NPREC.EQ.1) LPN=LPN+32
00145   31*    213 LP=LPN
00146   32*        IF (NLGF.LT.LGF) GO TO 205
00150   33*        DZ=DX
00151   34*        DX=1
00152   35*        I=1
00153   36*        J=0
00154   37*        CALL FIELD
00155   38*    250 NDX=DX+LGF*32
00156   39*        IF (LGF.EQ.0) NDX=DX+32
00160   40*        IF (LGF.EQ.0) GO TO 240
00162   41*        IX=DX
00163   42*        DX=DX+23
00164   43*        KD3=0
00165   44*        GO TO 216
00166   45*    214 I1=FLD(8,4,DATA(DX+3))
00167   46*        I2=FLD(2,4,DATA(DX+9))
00170   47*        KD3=I1+I2*10-1
00171   48*    215 DX=IX
00172   49*        IZ=IX+18
00173   50*    216 J=J+1
00174   51*        CALL FIELD3
00175   52*        IF (KD3.LE.0) GO TO 217
00177   53*        KD3=KD3-1
00200   54*        IF (KD3.EQ.0) GO TO 240
00202   55*        DX=DX+9
00203   56*        IF (DX.LE.IZ) GO TO 216
00205   57*    217 IX=IX+32
00206   58*        IF (KD3.GT.0) GO TO 215
00210   59*        IF (LGF.GT.1) GO TO 214
00212   60*        GO TO 240
00213   61*    220 LPN=LP+32
```

```
00214     62*          IF (LPN.LT.LMT) GO TO 221
00216     63*          IF (NPREC.EQ.IPREC) NPREC=0
00220     64*          CALL READ(5200)
00221     65*          LPN=LPN-KF
00222     66*          IF (NPREC.EQ.1) LPN=LPN+32
00224     67*      221 LP=LPN
00225     68*          GO TO 201
00226     69*      240 NO(I)=J
00227     70*          CMAX=I
00230     71*          IF (MFLAG.EQ.0) GO TO 199
00232     72*          IF (I.EQ.IPARTS) GO TO 199
00234     73*          I=I+1
00235     74*          J=0
00236     75*          DX=NDX
00237     76*          CALL FIELD2
00240     77*          GO TO 250
00241     78*      199 DX=DZ
00242     79*          RETURN
00243     80*      200 RETURN 1
00244     81*          END

       END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    0 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    1 SEC.
PHASE 6 TIME =    0 SEC.


TOTAL COMPILATION TIME =    1  SEC
```

```
WIT FOR FIELD,FIELD                                                                    09 JUN 71          15:22:45.573
UNIVAC 1108 FORTRAN V LEVEL   220b   S1, 150183
THIS COMPILATION WAS DONE ON 9 JUN 71 AT 15:22:45

   SUBROUTINE FIELD      ENTRY POINT 001362

             FIELD1     ENTRY POINT 001365


             FIELD3     ENTRY POINT 001370

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   001373
         0000   *DATA   000054
         0002   *BLANK  022646


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000247 1L          0001   001015 10L         0001   000251 2L          0001   000314 3L          0001   000316 4L
    0001   000362 5L          0001   000364 6L          0001   000430 7L          0001   000432 8L          0002 I 002313 A1A
    0002 I 002375 A1B         0002 I 002457 A3          0002 I 002472 A3A         0002 I 022645 A30LD       0002 I 002460 A4A
    0002 I 002461 A4B         0002 I 013240 B           0000 I 000002 BLANK       0002 I 002467 B10         0002 I 002470 B12A
    0002 I 002471 B13         0002 I 002473 B16         0002 I 022545 B16A        0002 I 002474 B17B        0002 I 002462 B3
    0002 I 002463 B5          0002 I 002464 B6          0002 I 002465 B8          0002 I 002466 B9          0002 I 013237 CFLAG
    0002 I 015750 CMAX        0002 I 002475 C1          0002 I 002557 C1A         0002 I 022546 C130        0002 I 022550 C5
    0002 I 002641 C7A1        0002 I 002723 C7A2        0002 I 003005 C7B1        0002 I 003067 C7B2        0002 I 003315 C9
    0002 I 017257 DATA        0002 I 016735 DELTA       0002 I 017256 DX          0002 I 003377 D1          0002 I 011267 D11
    0002 I 012253 D12         0002 I 013242 D13         0002 I 022547 D130        0002 I 004363 D3          0000 I 000013 D3A
    0000 I 000014 D3B         0000 I 000015 D3C         0002 I 005347 D4          0000 I 000016 D4A         0000 I 000017 D4B
    0000 I 000020 D4C         0002 I 001324 D4FRST      0002 I 006333 D           0002 I 007317 D8          0002 I 010303 D9
    0002 I 022635 GAMMA       0002 I 002310 I           0002 I 017247 IF          0002 I 022527 IHDR        0002 I 022363 INEED
    0002 I 017253 IPREX       0002 I 017252 IPREC       0002 I 017245 IDT         0002 I 017246 IS          0002 I 001000 ISTAT
    0000 I 000003 I1          0000 I 000004 I2          0000 I 000005 I3          0000 I 000006 I4          0000 I 000007 I5
    0000 I 000010 I6          0000 I 000011 I7          0000 I 000012 I8          0002 I 002311 J           0002 I 022357 KF
    0002 I 002312 LGF         0002 I 017250 LMT         0002 I 000000 LOAD        0002 I 017254 LP          0002 I 017255 LPN
    0002 I 015751 LXTIME      0002 I 022362 M           0002 I 022360 MFLAG       0002 I 022634 MINC1       0002 I 022361 N
    0000 I 000001 NA          0002 I 015130 NEED        0000 I 000000 NK          0002 I 022632 NN          0002 I 014226 NO
    0002 I 017251 NPREC       0002 I 001002 OPFAC       0002 R 003151 RC7A        0002 R 003233 RC7B        0002 I 014310 SAVEI
    0002 I 014620 SAVEJ       0002 I 001160 SOA3        0002 I 022633 STAND       0002 I 022640 S1          0002 I 022641 S2
    0002 I 022636 TF          0002 I 015440 TOS         0002 I 022637 TSM         0002 I 022643 X           0002 I 022644 Y
    0002 R 022642 ZMILE


00101      1*          SUBROUTINE FIELD
00103      2*          IMPLICIT INTEGER (A,B,C,D,F,G,O,S,T,X,Y)
00104      3*          COMMON LOAD(512),ISTAT(2),OPFAC(110),SOA3(100),D4FRST(50,10),I,J
00105      4*          COMMON LGF,A1A(50),A1B(50),A3,A4A,A4B,B3,B5,B6,B8,B9,B10,B12A,B13
```

```
00106     5*          COMMON A3A,B16,B173,C1(50),C1A(50),C7A1(50),C7A2(50),C7B1(50),
00106     6*         1 C7B2(50),RC7A(50),RC7B(50),C9(50),D1(50,10),D3(50,10),D4(50,10),
00106     7*         2 D5(50,10),D8(50,10),D9(50,10),D11(50,10),D12(50,10),CELAX,B(2),
00106     8*         3 D13(50,10),AD(50),SAVEI(200),SAVEJ(200),NEED(200),TOS(200),CMAX,
00106     9*         4 LVTIME(50,10),DELTA(200),IOT,IC,IF,LMT,NPREC,IPREC,INREX,LP,
00106    10*         5 LPH,DX,DATA(1600),KF,MFLAG,N,M,INFED(100),IHDR(14),B16A,C130,D130
00107    11*          COMMON CS(50), N,STAND,MINC1,GAMMA,TE,TSM,S1,S2,TITLE,X,Y,A30LD
00110    12*          DATA NK/'  ©©©©OK'/,NA/'©©©©NA'/,BLANK/'©©©©  '/
00114    13*          A3A=FLD(0,6,DATA(DX+2))
00115    14*          I1=FLD(8,4,DATA(DX+2))
00116    15*          I2=FLD(14,4,DATA(DX+2))
00117    16*          I3=FLD(20,4,DATA(DX+2))
00120    17*          I4=FLD(26,4,DATA(DX+2))
00121    18*          A3=I1*1000+I2*100+I3*10+I4
00122    19*          A30LD=I2*100+I3*10+I4
00123    20*          I1=FLD(32,4,DATA(DX+2))
00124    21*          I2=FLD(2,4,DATA(DX+3))
00125    22*          A4A=I1*10+I2
00126    23*          A4B=FLD(8,4,DATA(DX+3))
00127    24*          I1=FLD(26,4,DATA(DX+5))
00130    25*          I2=FLD(32,4,DATA(DX+5))
00131    26*          I3=FLD(2,4,DATA(DX+6))
00132    27*          I4=FLD(8,4,DATA(DX+6))
00133    28*          B3=I1*1000+I2*100 +I3*10+I4
00134    29*          IF (B3.GT.4095) B3=4095
00136    30*          I1=FLD(2,4,DATA(DX+7))
00137    31*          I2=FLD(8,4,DATA(DX+7))
00140    32*          I3=FLD(14,4,DATA(DX+7))
00141    33*          I4=FLD(20,4,DATA(DX+7))
00142    34*          I5=FLD(26,4,DATA(DX+7))
00143    35*          I6=FLD(32,4,DATA(DX+7))
00144    36*          I7=FLD(2,4,DATA(DX+8))
00145    37*          I8=FLD(8,4,DATA(DX+8))
00146    38*          B5=I1*10000000+I2*1000000+I3*100000+I4*10000+I5*1000+I6*100+I7*10
00146    39*         1 +I8
00147    40*          IF (B5.GT.130001) B5=130001
00151    41*          B6=FLD(12,12,DATA(DX+8))
00152    42*          IF (B6.EQ.NK.OR.B6.EQ.NA.OR.B6.EQ.BLANK) GO TO 1
00154    43*          I1=FLD(14,4,DATA(DX+8))
00155    44*          I2=FLD(20,4,DATA(DX+8))
00156    45*          B6=I1*10+I2
00157    46*          GO TO 2
00160    47*        1 B6=99
00161    48*        2 B8=FLD(0,12,DATA(DX+9))
00162    49*          IF (B8.EQ.NK.OR.B8.EQ.NA.OR.B8.EQ.BLANK) GO TO 3
00164    50*          I1=FLD(2,4,DATA(DX+9))
00165    51*          I2=FLD(8,4,DATA(DX+9))
00166    52*          B8=I1*10+I2
00167    53*          GO TO 4
00170    54*        3 B8=1
00171    55*        4 B9=FLD(12,12,DATA(DX+9))
00172    56*          IF (B9.EQ.NK.OR.B9.EQ.NA.OR.B9.EQ.BLANK) GO TO 5
00174    57*          I1=FLD(14,4,DATA(DX+9))
00175    58*          I2=FLD(20,4,DATA(DX+9))
00176    59*          B9=I1*10+I2
00177    60*          GO TO 6
00200    61*        5 B9=1
00201    62*        6 B10=FLD(24,12,DATA(DX+9))
```

```
00202    83*          IF (B10.EQ.NB. .B10.EQ.NA.OR.B10.EQ.BLANK) GO TO 7
00204    84*          I1=FLD(26,4,DATA(DX+9))
00205    85*          I2=FLD(32,4,DATA(DX+9))
00206    86*          B10=I1*10+I2
00207    87*          GO TO 8
00210    88*        7 B10=99
00211    89*        8 I1=FLD(8,4,DATA(DX+10))
00212    90*          I2=FLD(14,4,DATA(DX+10))
00213    91*          B12A=I1*10+I2
00214    92*          I1=FLD(26,4,DATA(DX+10))
00215    93*          I2=FLD(32,4,DATA(DX+10))
00216    94*          B13=I1*10+I2
00217    95*          B16A=FLD(18,6,DATA(DX+11))
00220    96*          I1=FLD(20,4,DATA(DX+11))
00221    97*          I2=FLD(26,4,DATA(DX+11))
00222    98*          I3=FLD(32,4,DATA(DX+11))
00223    99*          B16=I1*100+I2*10+I3
00224   100*          B17B=FLD(14,4,DATA(DX+12))
00225   101*          IF (MFLAG.EQ.1) DX=DX+32
00227   102*          ENTRY FIELD2
00230   103*          I1=FLD(8,4,DATA(DX))
00231   104*          I2=FLD(14,4,DATA(DX))
00232   105*          A1A(I)=I1*10+I2
00233   106*          I1=FLD(20,4,DATA(DX))
00234   107*          I2=FLD(26,4,DATA(DX))
00235   108*          I3=FLD(32,4,DATA(DX))
00236   109*          I4=FLD(2,4,DATA(DX+1))
00237   110*          I5=FLD(8,4,DATA(DX+1))
00240   111*          A1B(I)=I1*10000+I2*1000+I3*100+I4*10+I5
00241   112*          I1=FLD(14,4,DATA(DX+13))
00242   113*          I2=FLD(20,4,DATA(DX+13))
00243   114*          I3=FLD(26,4,DATA(DX+13))
00244   115*          I4=FLD(32,4,DATA(DX+13))
00245   116*          I5=FLD(2,4,DATA(DX+14))
00246   117*          I6=FLD(8,4,DATA(DX+14))
00247   118*          C1(I)=I1*100000+I2*10000+I3*1000+I4*100+I5*10+I6
00250   119*          C1A(I)=(I1*10+I2)*240+I3*100+I4*10+(I5*10+I6)/6
00251   120*          C5(I)=FLD(24,6,DATA(DX+15))
00252   121*          I1=FLD(32,4,DATA(DX+16))
00253   122*          I2=FLD(2,4,DATA(DX+17))
00254   123*          I3=FLD(8,4,DATA(DX+17))
00255   124*          I4=FLD(14,4,DATA(DX+17))
00256   125*          C7A1(I)=I1*10+I2
00257   126*          C7A2(I)=I3*10+I4
00260   127*          I1=FLD(26,4,DATA(DX+17))
00261   128*          I2=FLD(32,4,DATA(DX+17))
00262   129*          I3=FLD(2,4,DATA(DX+18))
00263   130*          I4=FLD(8,4,DATA(DX+18))
00264   131*          I5=FLD(14,4,DATA(DX+18))
00265   132*          C7B1(I)=I1*100+I2*10+I3
00266   133*          C7B2(I)=I4*10+I5
00267   134*          I1=FLD(8,4,DATA(DX+19))
00270   135*          I2=FLD(14,4,DATA(DX+19))
00271   136*          I3=FLD(20,4,DATA(DX+19))
00272   137*          I4=FLD(26,4,DATA(DX+19))
00273   138*          C9(I)=I1*1000+I2*100+I3*10+I4
00274   139*          LGF=FLD(2,4,DATA(DX))
00275   140*          RETURN
```

```
00275   121*          ENTRY FIELD3
00277   122*          IF (FLD(12,24, DATA(DX)).NE.0) GO TO 10
00301   123*          J=J-1
00302   124*          RETURN
00303   125*       10 I1=FLD(14,4,DATA(DX))
00304   126*          I2=FLD(20,4,DATA(DX))
00305   127*          D1(I,J)=I1*10+I2
00306   128*          I1=FLD(26,4,DATA(DX+1))
00307   129*          I2=FLD(32,4,DATA(DX+1))
00310   130*          D3A=I1*10+I2
00311   131*          I1=FLD(2,4,DATA(DX+2))
00312   132*          I2=FLD(8,4,DATA(DX+2))
00313   133*          D3B=I1*10+I2
00314   134*          I1=FLD(14,4,DATA(DX+2))
00315   135*          I2=FLD(20,4,DATA(DX+2))
00316   136*          D3C=I1*10+I2
00317   137*          D3(I,J)=D3A*24 +D3B*10+D3C/6
00320   138*          IF (D3(I,J).EQ.0) D3(I,J)=C1A(I)
00322   139*          I1=FLD(26,4,DATA(DX+2))
00323   140*          I2=FLD(32,4,DATA(DX+2))
00324   141*          D4A=I1*10+I2
00325   142*          I1=FLD(2,4,DATA(DX+3))
00326   143*          I2=FLD(8,4,DATA(DX+3))
00327   144*          D4B=I1*10+I2
00330   145*          I1=FLD(14,4,DATA(DX+3))
00331   146*          I2=FLD(20,4,DATA(DX+3))
00332   147*          D4C=I1*10+I2
00333   148*          D4(I,J)=D4A*240+D4B*10+D4C/6
00334   149*          IF (D4(I,J).EQ.0) D4(I,J)=D3(I,J)
00336   150*          I1=FLD(14,4,DATA(DX+4))
00337   151*          I2=FLD(20,4,DATA(DX+4))
00340   152*          I3=FLD(26,4,DATA(DX+4))
00341   153*          I4=FLD(32,4,DATA(DX+4))
00342   154*          D6(I,J)=I1*1000+I2*100+I3*10+I4
00343   155*          I1=FLD(32,4,DATA(DX+5))
00344   156*          I2=FLD(2,4,DATA(DX+6))
00345   157*          I3=FLD(8,4,DATA(DX+6))
00346   158*          I4=FLD(14,4,DATA(DX+6))
00347   159*          D8(I,J)=I1*1000+I2*100+I3*10+I4
00350   160*          I1=FLD(20,4,DATA(DX+6))
00351   161*          I2=FLD(26,4,DATA(DX+6))
00352   162*          D9(I,J)=I1*10+I2
00353   163*          I1=FLD(8,4,DATA(DX+7))
00354   164*          I2=FLD(14,4,DATA(DX+7))
00355   165*          D11(I,J)=I1*10+I2
00356   166*          I1=FLD(20,4,DATA(DX+7))
00357   167*          I2=FLD(26,4,DATA(DX+7))
00360   168*          D12(I,J)=I1*10+I2
00361   169*          I1=FLD(32,4,DATA(DX+7))
00362   170*          I2=FLD(2,4,DATA(DX+8))
00363   171*          D13(I,J)=I1*10+I2
00364   172*          RETURN
00365   173*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    1 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    1 SEC.
```

```
.I ASM RAND,RAND
ASSEMBLED BY UNIVAC 1108 EXEC II ASSEMBLER   2404 00 RA
THIS ASSEMBLY WAS DONE 0. 08 JUN 71 AT 15:22:46
000001      00   000000  71 13 01 13 1 000001    RAND*  DL     13,*1,11
000002           000001  71 12 01 00 0 000000           DS     13,IS
000003           000002  27 17 14 00 0 000000           L,017  12,0
000004           000003  27 00 15 00 0 000001           L      13,IS+1
000005           000004  32 00 01 00 0 000002           MF     13,M
000006           000005  73 00 02 00 0 000001           SSC    14,1
000007           000006  27 00 17 00 0 000000           L      15,IS
000008           000007  30 00 03 00 0 000002           MI     15,M
000009           000010  71 10 00 00 0 000017           DA     12,15
000010           000011  73 12 01 00 0 000001           LSSL   13,1
000011           000012  73 02 01 00 0 000001           SSL    13,1
000012           000013  71 12 01 00 0 000000           DS     13,IS
000013           000014  27 17 14 00 0 000170           L,017  12,0170
000014           000015  76 05 00 00 0 000015           LCF    12,13
000015           000016  27 00 14 00 0 000015           L      12,13
000016           000017  71 13 01 00 0 000000           DL     13,IS
000017           000020  71 12 01 13 1 000001           DS     13,*1,11
000018           000021  51 00 00 13 1 000000           TNZ    *0,11
000019           000022  74 04 00 13 0 000003            J      3,11
000020           000023  24 00 14 00 0 000003           A      12,BIT
000021           000024  76 01 00 00 0 000004           FAN    12,ONE
000022           000025  74 04 00 13 0 000003           J      3,11
000023      01   000000  154447730601          $(1),IS  +0154447730601
000024           000001  255751305264                   +0255751305264
000025           000002  011060471625          M        +0011060471625
000026           000003  001000000000          BIT      +0001000000000
000027           000004  200777777777          ONE      +0200777777777
000028                                                  END
```

```
@II ASM 10,10                                                                                    08 JUN 71        18:08:22.743
ASSEMBLED BY UNIVAC 1108 EXEC II ASSEMBLER  2404 0008A
THIS ASSEMBLY WAS DONE ON 08 JUN 71 AT 18:08:22
000001      00   000000  000000000003                    +3
000002           000001  000000000002                    +2
000003           000002  000000000001                    +1
000004           000003  777777777773          NTAB$*   -4
000005           000004  000000000001                    +1
000006           000005  000000000002                    +2
000007           000006  000000000003                    +3
000008           000007  000000000004                    +4
000009           000010  000000000001                    +1
000010           000011  000000000002                    +2
000011           000012  000000000006                    +'A'
000012           000013  000000000007                    +'B'
000013           000014  000000000010                    +'C'
000014           000015  000000000031                    +DRUM1
000015           000016  000000000037                    +DRUM2
000016           000017  000000000045                    +DRUM3
000017           000020  000000000053                    +DRUM4
000018           000021  000000000061                    +DRUM5
000019           000022  000000000067                    +DRUM6
000020           000023  000000000075                    +DRUM7
000021           000024  000000000103                    +DRUM8
000022           000025  000000000111                    +DRUM9
000023           000026  000000000117                    +DRUM10
000024           000027  000000000125                    +DRUM11
000025           000030  000000000133                    +DRUM12
000026           000031  000001300000          DRUM1    +360448
000027           000032  000001603237                    +460447
000028                                                 DO 4 , +0
                 000033  000000000000
                 000034  000000000000
                 000035  000000000000
                 000036  000000000000
000029           000037  000001603240          DRUM2    +460448
000030           000040  000001721337                    +500447
000031                                                 DO 4 , +0
                 000041  000000000000
                 000042  000000000000
                 000043  000000000000
                 000044  000000000000
000032           000045  000001721340          DRUM3    +500448
000033           000046  000002250217                    +610447
000034                                                 DO 4 , +0
                 000047  000000000000
                 000050  000000000000
                 000051  000000000000
                 000052  000000000000
000035           000053  000002250220          DRUM4    +610448
000036           000054  000002342677                    +640447
000037                                                 DO 4 , +0
                 000055  000000000000
                 000056  000000000000
                 000057  000000000000
                 000060  000000000000
000038           000061  000002342700          DRUM5    +640448
000039           000062  000002646137                    +740447
```

000040				DO 4 , +0
	000053	[illegible]		
	000054	[illegible]		
	000055	[illegible]		
	000056	[illegible]		
000041	000057	000002745140	[illegible]	+740448
000042	000070	000002764237		+750447
000043				DO 4 , +0
	000071	000000000000		
	000072	000000000000		
	000073	000000000000		
	000074	000000000000		
000044	000075	000002764240	DRUM7	+780448
000045	000076	000003125757		+830447
000046				DO 4 , +0
	000077	000000000000		
	000100	000000000000		
	000101	000000000000		
	000102	000000000000		
000047	000103	000003125760	DRUM8	+830448
000048	000104	000003220437		+860447
000049				DO 4 , +0
	000105	000000000000		
	000106	000000000000		
	000107	000000000000		
	000110	000000000000		
000050	000111	000003220440	DRUM9	+860448
000051	000112	000003232247		+865447
000052				DO 4 , +0
	000113	000000000000		
	000114	000000000000		
	000115	000000000000		
	000116	000000000000		
000053	000117	000003232250	DRUV10	+865448
000054	000120	000003313117		+890447
000055				DO 4 , +0
	000121	000000000000		
	000122	000000000000		
	000123	000000000000		
	000124	000000000000		
000056	000125	000003313120	DRUM11	+890448
000057	000126	000003373767		+915447
000058				DO 4 , +0
	000127	000000000000		
	000130	000000000000		
	000131	000000000000		
	000132	000000000000		
000059	000133	000003373770	DRUM12	+915448
000060	000134	000003454637		+940447
000061				DO 4 , +0
	000135	000000000000		
	000136	000000000000		
	000137	000000000000		
	000140	000000000000		
000062				END

```
@IT FOR DEMGEN,DEMGEN                                                    08 JUN 71          18:08:23.703
UNIVAC 1108 FORTRAN V LEVEL  2206 0018 F50142
THIS COMPILATION WAS DONE ON 08 JUN 71 AT 18:08:23

  MAIN PROGRAM

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   003352
        0000   *DATA   005176
        0002   *BLANK  116563
        0003   BIX     002733
        0004   AIX     000367
        0005   SEAL    000001


  EXTERNAL REFERENCES (BLOCK, NAME)

        0006   MELVIN
        0007   NTRAN
        0010   SELECT
        0011   BOX
        0012   ADD
        0013   RAND
        0014   ADTIME
        0015   EXIT
        0016   NRDJ$
        0017   NIO1$
        0020   NIO2$
        0021   NWDJ$
        0022   NRBJ$
        0023   NSTOP$
        0024   ALOG


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   002703 100L      0001   003127 101L      0001   002454 102L      0001   001541 1021G     0001   002456 103L
   0001   001550 1030G     0001   001555 1036G     0001   002530 104L      0001   001605 1055G     0001   002415 106L
   0001   001617 1064G     0001   002525 107L      0001   001637 1075G     0001   002607 110L      0001   002535 111L
   0001   001720 1113G     0001   002606 114L      0001   002204 115L      0001   001745 116L      0001   002230 118L
   0001   000012 120G      0001   002266 1217G     0001   000024 127G      0001   002430 1274G     0001   002515 1313G
   0001   002537 1326G     0001   002555 1340G     0001   002572 1350G     0001   002573 1353G     0001   002616 1366G
   0001   002645 1402G     0001   002646 1404G     0001   002743 1447G     0001   002761 1461G     0001   002773 1467G
   0001   003005 1475G     0001   001075 15L       0001   003046 1511G     0001   003062 1521G     0001   000074 153G
   0001   003156 1566G     0001   003174 1575G     0001   003207 1603G     0001   003217 1612G     0001   003232 1620G
   0001   003245 1631G     0001   003251 1635G     0001   000116 164G      0001   003272 1645G     0001   003304 1656G
   0001   003320 1665G     0001   003326 1672G     0001   003340 1702G     0001   000140 173G      0001   000140 176G
   0001   000056 20L       0001   000145 202G      0001   001533 21L       0001   002270 222L      0001   000174 224G
   0001   002304 224L      0001   000212 235G      0001   000224 244G      0001   000224 246G      0001   000667 25L
   0001   000224 250G      0001   003276 255L      0001   003330 260L      0001   000245 266G      0001   002677 27L
   0001   002413 28L       0001   001437 29L       0001   000265 301G      0001   000270 304G      0001   000312 312G
   0001   000321 320G      0001   000406 346G      0001   000417 355G      0001   000426 363G      0001   000442 372G
   0001   000375 4L        0001   000453 402G      0001   000464 412G      0001   000542 442G      0001   000600 456G
   0001   000647 467G      0000   004561 500F      0000   004601 501F      0000   004603 502F      0000   004604 503F
```

```
0000   004607 5[illegible]F     0[illegible]   004[illegible]1 [illegible]5F     0000   005505 5[illegible]F     0000   004637 [illegible]7F     0001   000702 507G
0000   004612 509F     0[illegible]   004[illegible]5  10F     0000   004574 511F     0001   004575 512F     0000   004576 516F
0001   000751 526G     0001   000754 [illegible]     0001   000755 5[illegible]G     0001   004577 5[illegible]0F     0000   004660 551F
0000   004752 552F     0001   000770 [illegible]526     0000   004764 555F     0000   004710 555F     0000   004645 560F
0001   001016 564G     0001   001031 [illegible]56G     0000   004712 5[illegible]F     0000   004766 5[illegible]1F     0000   004715 585F
0000   004771 586F     0000   004774 [illegible]0F     0000   004775 591F     0000   005012 5[illegible]2F     0000   004721 595F
0000   005000 596F     0001   000350 [illegible]L     0001   000554 60[illegible]     0001   001053 602G     0001   001062 605G
0000   004614 609F     0000   004562 610F     0000   004547 61[illegible]F     0001   001105 615G     0000   004701 620F
0001   001112 621G     0000   004726 [illegible]25F     0001   001004 63L     0000   004740 630F     0001   001124 630G
0000   005003 635F     0001   001144 637G     0000   004754 64[illegible]F     0001   001146 643G     0001   001160 652G
0001   001153 657G     0001   001211 [illegible]75G     0001   001217 703G     0001   001305 742G     0001   002115 75L
0001   001351 755G     0001   002175 76L     0001   000457 8L     0001   001262 80L     0001   001361 81L
0001   002315 82L     0001   002242 [illegible]3L     0001   000721 85L     0001   001271 89L     0001   001363 89L
0001   000546 9L     0001   002354 90L     0001   003213 900L     0000   004627 9000F     0000   005014 9001F
0000   005070 9002F     0000   005100 9003F     0000   004630 9004F     0000   004662 9005F     0001   002347 91L
0001   002344 92L     0001   002726 95L     0001   003115 98L     0000 I 002761 AGRAGA     0004 R 000360 B
0003 I 000000 CAL     0000 I 002623 CAS2DA     0002 R 116161 CBOX     0002 R 116131 CPER     0000 R 003120 DAYS
0004 R 000356 DEL     0000 R 004422 DELTA     0000 R 004551 DELTF     0000 I 004509 I     0002 I 000001 IADR
0000 I 004521 IBOX     0002 I 116171 ICASE     0004 I 000364 IDAY     0004 I 000020 IDELTA     0000 I 004552 IDELTF
0000 I 004527 IDNO     0003 I 002725 IEWPP     0004 I 000021 IHOL     0000 I 004530 IHOUR     0000 I 004544 IL
0004 I 000045 ILIST     0000 I 004531 IMIN     0000 I 004541 INUM     0000 I 004532 INUMB     0000 I 004557 INUMBA
0004 I 000356 ISCED     0004 I 000357 ISCEH     0000 I 004525 ISCEN     0000 I 004507 ISCSD     0000 I 004506 ISCSH
0004 I 000000 ISWIT     0003 I 002720 ISWPP     0004 I 000355 ITIME     0000 I 004536 IT1     0000 I 004515 IUNIT
0000 I 004513 IWK     0000 I 004517 IWRITE     0000 I 004503 IYEARS     0000 I 004504 IYR     0000 I 003117 I$
0000 I 004533 I1     0000 I 004534 I2     0000 I 004535 I3     0000 I 004510 J     0000 I 004555 JDAY
0000 I 004560 JJ     0000 I 004526 JTIME     0000 I 004511 K     0000 I 004514 KK     0000 I 004545 KL
0000 I 004523 L     0004 R 000055 LAMBDA     0000 I 004477 LSTSTN     0000 I 004524 M     0000 I 004502 MAXCA
0000 I 004505 MO     0002 I 106241 NADR     0000 I 004542 NBLOCK     0000 I 003130 NC     0000 I 004546 NCAS
0000 I 004240 NCASE     0002 I 116101 NCASES     0000 I 004516 NCASIN     0000 I 004556 NC2D     0000 I 004550 NDAY
0000 I 004270 NEED     0000 I 003440 NEWCAS     0000 I 004520 NEXTRA     0002 I 000000 NODRUM     0004 I 000363 NOHOLS
0003 I 002732 NPP     0000 I 004547 NPRIOR     0000 I 004512 NSCEN     0000 I 004543 NSPACE     0004 I 000362 NSPP
0004 I 000015 NSPPE     0004 I 000012 NSPPS     0000 I 004522 NT     0005 I 000000 NUPPER     0000 R 004501 PER
0003 R 000000 RAND     0000 R 004540 RN     0000 I 002621 STATN     0000 I 003074 TEMP     0004 R 000365 TIME
0000 R 004554 TMIN     0000 R 004345 TOS     0000 I 002736 TOTAL     0000 I 002622 TTOS     0000 I 000000 VLDMTX
0000 R 004537 XLAMB     0000 R 003740 XNLAMB     0000 R 004553 ZMIN     0000 R 003430 ZNDAYS
```

```
00101     1*          PARAMETER NP=24,NN=45,MXBX=4500,NXBX=1000,NH=20,NCP=250,NS=75      *WS 2/7
00101     2*        1          ,NV=19,NA=48                                              *WS 2/71
00103     3*         INTEGER VLDMTX(NS,NV),STATN,TTOS,CAS2DA(NS),TOTAL(NV),AGRAGB(NS)   *WS 2/71
00103     4*        1        ,CAL(12,31,4),TEMP(NV)
00104     5*         REAL LAMBDA(8,NP)
00105     6*         DATA AGRAGB /9,9,NA,9,44,44,44,32,32,9,9,9,NA,18,19,NA,NA,17,26,20*WS 2/71
00105     7*        1     ,13,15,34,29,NA,NA,NA,NA,NA,2,NA,2,27,37,1,NA,4,31,3,NA        *WS 2/71
00105     8*        2     ,16,23,28,6,42,25,47,45,35,40,46,10,30,43,33,12,11,39,38,NA    *WS 2/71
00105     9*        3     ,41,14,36,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA/                 *WS 2/71
00107    10*         DIMENSION ISWIT(10),IHOL(NH),ILIST(2,2,2),DAYS(8),CBOX(8),NC(8,NP)
00107    11*        1,IADR(8,MXBX),NADR(4,NXBX),CPER(NP),NCASES(NP),ICASE(NCP)
00107    12*        2          ,NSPPS(3),NSPPE(3)
00107    13*        3          ,B(2)
00107    14*        4          ,ZNDAYS(8),NEWCAS(8,NP),XNLAMB(8,NP)
00107    15*        5          ,NCASE(NP),NEED(NN),TOS(NN),DELTA(NN)
00110    16*         COMMON/BIX/              CAL,ISWPP(5),IEWPP(5),NPP
00111    17*         COMMON/AIX/ ISWIT,NSPPS,NSPPE,IDELTA,IHOL,ILIST,LAMBDA,ITIME,
00111    18*        1ISCED,ISCEH,B,NSPP,NOHOLS,IDAY,TIME,DEL
00112    19*         COMMON/SEAL/ NUPPER
```

```
00113   20*          COMMON NODRUM,IADR,MADR,NCASES,CPER,CDOY,ICASE
00114   21*          CALL MELVIN
00115   22*          LSTSTN=1                                                    *WS 2/71
00115   23*    C
00115   24*    C     READ  IN PARAMETERS DEFINING THE RUN
00115   25*    C     ISWIT(1)=1--ISWIT(2)=0---CASELOAD SAME AS THAT OF INPUT TAPE
00115   26*    C     ISWIT(2)=1 --CASELOAD CHANGES BY 'PER' PERCENT
00115   27*    C     ISWIT(1)=3 --CASES ARE INCREASED BY OPFAC AND CASE PARAMETER CRITERIA
00115   28*    C
00116   29*          READ(5,500) (ISWIT(I),I=1,10)
00124   30*      500 FORMAT(10I1)
00125   31*          WRITE(6,610) (ISWIT(I),I=1,10)
00133   32*      610 FORMAT(1H1,20X,'THE VALUES OF THE SWITCH ARRAY ARE'/1X,10I5)
00134   33*          IF(ISWIT(2).EQ.1)READ(5,511)PER                             *WS 2/71
00140   34*      511 FORMAT(F5.0)
00141   35*          PER=PER/100.
00142   36*          IF(ISWIT(10).EQ.1)  READ(5,512)  MAXCA
00146   37*      512 FORMAT(I5)
00147   38*          IF(ISWIT(1).EQ.3) READ(5,516) NODRUM,(CPER(I),I=1,NODRUM)
00157   39*      516 FORMAT(I2,12(F5.2))
00157   40*    C
00157   41*    C     READ IN NO. OF PEAK PERIODS AND NO. OF YEARS IN CALENDAR
00157   42*    C     START AND END WEEK OF EACH PEAK PERIOD
00157   43*    C
00160   44*          READ(5,501) NPP,IYEARS,(ISWPP(I),IEWPP(I),I=1,NPP)
00171   45*      501 FORMAT(2I2,20I3)
00172   46*          DO 1 IYR=1,IYEARS
00175   47*          DO 1 IDAY =1,31
00200   48*          READ(5,502) (CAL(MO,IDAY,IYR),MO=1,12)
00206   49*        1 CONTINUE
00211   50*      502 FORMAT(12I6)
00211   51*    C
00211   52*    C     READ SCENARIO LIMITS START HOUR START DAY END HOUR END DAY
00211   53*    C
00212   54*          READ(5,503) ISCSH,ISCSD,ISCEH,ISCED
00220   55*      503 FORMAT(6I4)
00221   56*          READ(5,506) NSPP, (NSPPS(I),NSPPE(I),I=1,NSPP)
00231   57*      506 FORMAT(I2,(10I3))
00231   58*    C
00231   59*    C     READ TABLE OF HOLIDAYS
00231   60*    C
00232   61*          READ(5,504) NOHOLS,(IHOL(I),I=1,NOHOLS)
00241   62*      504 FORMAT(I2,20I3)
00241   63*    C
00241   64*    C     READ BOX IDENTIFICATION TABLE
00241   65*    C
00242   66*          READ(5,505)  (((ILIST(I,J,K),I=1,2),J=1,2),K=1,2)
00256   67*      505 FORMAT(8I1)
00256   68*    C
00256   69*    C     READ RANDOM NUMBER SEEDS
00256   70*    C
00257   71*          READ(5,509) B(1),B(2)
00263   72*      509 FORMAT(O12,1X,O12)
00264   73*          WRITE(6,609)B                                               *WS 2/71
00272   74*      609 FORMAT('  THE ORIGINAL RANDOM NUMBER SEED WAS ',O12,O12)     *WS 2/71
00273   75*          READ(5,512)NSCEN                                            *WS 3/71
00273   76*    C
00273   77*    C     CALCULATE NO. OF DAYS IN EACH BOX USING THE CALENDAR
```

```
00273    78*    C
00275    79*          IF(ISWIT(3).EQ.1) GO TO 63
00300    80*          DO 2 I=1,NDAYS
00303    81*          DO 3 MO=1,12
00306    82*          K=MO+6
00307    83*          IF(K.GT.12) K=MO-6
00311    84*          DO 4 IDAY =1,31
00314    85*          IWK=CAL(K,IDAY,I)
00315    86*          IF(IWK.EQ.0) GO TO 4
00317    87*          DO 5 KK=1,NPP
00322    88*          IF(IWK.GE.ISWPP(KK).AND.IWK.LE.IFWPP(KK)) GO TO 6
00324    89*        5 CONTINUE
00326    90*          J=5
00327    91*          IF(MOD(IWK,2).NE.0) J=7
00331    92*          DAYS(J)=DAYS(J)+1
00332    93*          GO TO 4
00333    94*        6 J=1
00334    95*          IF(MOD(IWK,2).NE.0) J=3
00336    96*          DAYS(J)=DAYS(J)+1
00337    97*        4 CONTINUE
00341    98*        3 CONTINUE
00343    99*        2 CONTINUE
00345   100*          DO 7 I=2,8,2
00350   101*          DAYS(I)=DAYS(I-1)
00351   102*        7 CONTINUE
00353   103*          WRITE(6,510)  (DAYS(I),I=1,8)
00361   104*      510 FORMAT(2X,8F5.0)
00361   105*    C
00361   106*    C     INITIALIZE COUNTERS
00361   107*    C
00362   108*          DO 10 I=1,8
00365   109*          CBOX(I)=0.
00366   110*          IUNIT=I+9
00367   111*          CALL NTRAN(IUNIT,10)
00370   112*          IADR(I,1)=1
00371   113*          DO 11 J=1,24
00374   114*          NC(I,J)=0
00375   115*       11 CONTINUE
00377   116*       10 CONTINUE
00401   117*          DO 18 I=1,4
00404   118*          NADR(I,1)=1
00405   119*       18 CONTINUE
00405   120*    C
00405   121*    C     READ IN FIRST 28 CASE PARAMETERS
00407   122*          NCASIN=0
00410   123*        8 READ(7) (ICASE(I),I=1,28)
00416   124*          NCASIN=NCASIN+1
00417   125*          IWRITE=1
00420   126*          IF(ICASE(1).EQ.99)  GO TO 35
00422   127*          NEXTRA=(ICASE(15)-1)*3
00423   128*          IF(ICASE(15).EQ.0) NEXTRA=0
00425   129*          NUPPER=29+NEXTRA-1
00426   130*          IF(ICASE(15).GE.35)WRITE(6,9000)NCASIN,NUPPER,ICASE(1),ICASE(15)
00435   131*     9000 FORMAT(4I10)
00436   132*          IF(NEXTRA.EQ.0)  GO TO 9
00436   133*    C ***************************************************************
00436   134*    C     CASE CONTAINS MORE THAN 28 WORDS READ IN REMAINDER
00436   135*    C ***************************************************************
```

```
00440   136*          READ (7) (ICASE(I),I=2,NUPPER)
00446   137*        1 IF(ISWIT(1).EQ.1)  GO TO 60
00450   138*          CALL SELECT
00451   139*       60 IWRITE=1
00451   140*    C ******************************************************************
00451   141*    C     SUBROUTINE BOX DETERMINES BOX AND HOUR FOR OCCURANCE TIME OF CASE
00451   142*    C ******************************************************************
00452   143*       50 CALL BOX(IBOX,NT)
00452   144*    C ******************************************************************
00452   145*    C     INCREMENT CASE COUNTER FOR THE BOX AND HOUR
00452   146*    C ******************************************************************
00453   147*          NC(IBOX,NT)=NC(IBOX,NT)+1
00454   148*          IUNIT=IBOX+9
00455   149*          DO 62 I=1,IWRITE
00455   150*    C ******************************************************************
00455   151*    C     INCREMENT BOX CASE COUNTER
00455   152*    C ******************************************************************
00460   153*          CBOX(IBOX)=CBOX(IBOX)+1.
00461   154*          K=CBOX(IBOX)
00461   155*    C ******************************************************************
00461   156*    C ******************************************************************
00461   157*    C     DETERMINE STARTING DRUM ADRESS FOR NEXT CASE IN THIS BOX
00461   158*    C ******************************************************************
00462   159*          IADR(IBOX,K+1)=IADR(IBOX,K)+NUPPER
00462   160*    C ******************************************************************
00462   161*    C     WRITE CASE ONTO DRUM IN ITS APPROPRIATE BOX
00462   162*    C ******************************************************************
00463   163*          CALL NTRAN(IUNIT,1,NUPPER,ICASE(1),L)
00464   164*          IF(ICASE(15).LE.0)WRITE(6,9004)(ICASE(M),M=1,NUPPER)              *WS 2/71
00473   165*     9004 FORMAT(/'  BAD CASE -- 0 NEEDS '/(10I10))                         *WS 2/71
00474   166*       20 IF(L+1) 25,20,
00477   167*       62 CONTINUE
00501   168*          GO TO 8
00502   169*       25 WRITE(6,507)
00504   170*      507 FORMAT(2X,'ERROR IN NTRAN TRANSMISSION')
00505   171*          WRITE(6,555) (CBOX(I),I=1,8)
00513   172*          WRITE(6,560) IUNIT,L,IBOX,NT
00521   173*      560 FORMAT(1X,4I6)
00522   174*          STOP
00523   175*       85 IF(ISWIT(1).EQ.3) CALL ADD
00523   176*    C ******************************************************************
00523   177*    C     REWIND ALL DRUM UNITS
00523   178*    C ******************************************************************
00525   179*          DO 73 I=1,8
00530   180*          IUNIT=I+9
00531   181*          CALL NTRAN(IUNIT,10)
00532   182*       73 CONTINUE
00532   183*    C ******************************************************************
00532   184*    C     COMPUTE HISTORICAL LAMBDAS FOR ALL BOXES AND TIMES
00532   185*    C ******************************************************************
00534   186*          WRITE(6,615)
00536   187*      615 FORMAT(// 20X,'HISTORICAL MEAN HOURLY ARRIVAL RATES'//)
00537   188*          DO 86 I=1,8
00542   189*          DO 87 J=1,24
00545   190*          LAMBDA(I,J)=NC(I,J)/DAYS(I)
00546   191*       87 CONTINUE
00550   192*          WRITE(6,550)  (LAMBDA(I,J),J=1,24)
00556   193*       86 CONTINUE
```

```
00560   194*      63 CONTINUE
00561   195*         IF (ISWIT(3).EQ.1) READ(5,551) ((LAMBDA(I,J),J=1,24),I=1,8)
00573   196*         IF(ISWIT(3).EQ.1) WRITE(6,615)
00576   197*     551 FORMAT(12F5.3)
00577   198*         IF(ISWIT(2).EQ.0)GOTO15                                    *WS 2/71
00577   199*   C ***********************************************************
00577   200*   C     HISTORICAL CASE LOAD IS TO BE INCREASED BY PER PERCENT
00577   201*   C     THIS IS EQUIVILENT TO MULTIPLYING THE LAMBDAS BY (1+PER)
00601   202*         DO 16 I=1,8
00601   203*   C ***********************************************************
00604   204*         DO 17 J=1,24
00607   205*         LAMBDA(I,J)=LAMBDA(I,J)*(1.+PER)
00610   206*      17 CONTINUE
00612   207*      16 CONTINUE
00614   208*      15 DO 300 I=1,8
00617   209*         WRITE(6,550)  (LAMBDA(I,J),J=1,24)
00625   210*     300 CONTINUE
00627   211*         DO 1000 ISCEN=1,NSCEN                                      *WS 3/71
00632   212*         WRITE(6,9005)ISCEN                                         *WS 3/71
00635   213*    9005 FORMAT('1'///25X,'THE FOLLOWING IS VERSION',I3,' OF THIS SCENARIO.*WS 3/71
00635   214*        1'/////)                                                    *WS 3/71
00636   215*         DO321IBOX=1,8                                              *WS 3/71
00641   216*         ZNDAYS(IBOX)=0                                             *WS 3/71
00642   217*         DO320JTIME=1,NP                                            *WS 3/71
00645   218*     320 NEWCAS(IBOX,JTIME)=0                                       *WS 3/71
00647   219*     321 CONTINUE                                                   *WS 3/71
00651   220*         DO330I=1,NV                                                *WS 3/71
00654   221*         TOTAL(I)=0                                                 *WS 3/71
00655   222*         TEMP(I)=0                                                  *WS 3/71
00656   223*         DO329STATN=1,LSTSTN                                        *WS 3/71
00661   224*     329 VLDMTX(STATN,I)=0                                          *WS 3/71
00663   225*     330 CONTINUE                                                   *WS 3/71
00665   226*         IF(ISWIT(3).EQ.1) GO TO 80
00667   227*     550 FORMAT(1X,24F5.2)
00670   228*         WRITE (6,620)
00672   229*     620 FORMAT(//1X,'NUMBER OF CASES IN EACH BOX'//)
00673   230*         WRITE(6,555) (CBOX(I),I=1,8)
00701   231*     555 FORMAT(1X,8F7.0)
00702   232*         DO 72 I=1,8
00705   233*         IUNIT=I+9
00706   234*         CALL NTRAN(IUNIT,10)
00707   235*      72 CONTINUE
00711   236*         IDNO=3
00712   237*         IDAY=0
00713   238*         IHOUR=0
00714   239*         IMIN=0
00715   240*         WRITE(8,580)  IDNO,IDAY,IHOUR,IMIN
00723   241*         WRITE(6,580)  IDNO,IDAY,IHOUR,IMIN                         *WS 4/71
00731   242*     580 FORMAT(I3,I4,I3,I2)
00732   243*         IDNO=1
00733   244*      80 ITIME=ISCSH
00734   245*         TIME=ISCSH
00735   246*         IDAY=ISCSJ
00736   247*         INJMB=0
00737   248*      88 INJMB=INJMB+1
00740   249*         I1=2
00741   250*         DO 113 J=1,NSPP
00744   251*         IF(IDAY.GE.NSPPS(J).AND.IDAY.LE.NSPPE(J)) I1=1
```

```
00746   252*      113 CONTINUE
00750   253*       79 I2=1
00751   254*          K=MOD(IDAY,7)
00752   255*          IF(K.EQ.0.OR.K.EQ.6) I2=2
00754   256*          DO 84 J=1,NOHOLS
00757   257*          IF(IDAY.EQ.IHOL(J)) GO TO 81
00761   258*       84 CONTINUE
00763   259*          GO TO 89
00764   260*       81 I2=2
00765   261*       89 I3=1
00766   262*          IF(ITIME.LT.0800.OR.ITIME.GE.2000) I3=2
00770   263*          IBOX=ILIST(I1,I2,I3)
00771   264*          IF(ISWIT(3).EQ.1) GO TO 29
00773   265*          IT1=CBOX(IBOX)
00774   266*          IF(IT1.EQ.0) DEL=60.
00776   267*          IF(IT1.EQ.0) GO TO 75
01000   268*       29 IUNIT=IBOX+9
01001   269*          CALL NTRAN(IUNIT,10)
01002   270*          JTIME=(ITIME/100)+1
01003   271*          NEWCAS(IBOX,JTIME)=NEWCAS(IBOX,JTIME)+1
01004   272*          XLAMB=LAMBDA(IBOX,JTIME)
01005   273*          RN=RAND(0,B)
01006   274*          IF(ISWIT(3).EQ.1) GO TO 116
01010   275*          INUM=CBOX(IBOX)*RN+1
01011   276*          NBLOCK=IADR(IBOX,INUM+1)-IADR(IBOX,INUM)
01012   277*          NSPACE=IADR(IBOX,INUM)-1
01013   278*          CALL NTRAN(IUNIT,6,NSPACE)
01014   279*          CALL NTRAN(IUNIT,2,NBLOCK,ICASE(1),L)
01015   280*       21 IF(L+1) 25,21,
01020   281*          DO 94 I=1,3
01023   282*          NCASE(I)=ICASE(I)
01024   283*       94 CONTINUE
01026   284*          NCASE(4)=ICASE(6)
01027   285*          DO 96 I=5,15
01032   286*          NCASE(I)=ICASE(I+3)
01033   287*       96 CONTINUE
01035   288*          DO 97 I=16,21
01040   289*          NCASE(I)=ICASE(I+4)
01041   290*       97 CONTINUE
01043   291*          NCASE(22)=IBOX
01044   292*          NCASE(23)=ICASE(7)
01045   293*          IHOUR=ITIME/100
01046   294*          IMIN=ITIME-(IHOUR*100)
01047   295*          WRITE(8,585)  IDNO,IDAY,IHOUR,IMIN,(NCASE(I),I=1,12)
01061   296*      585 FORMAT(I3,I4,I3,I2,I3,11I5)
01062   297*          WRITE(8,590)  (NCASE(I),I=13,23)
01070   298*          IF(ICASE(15).EQ.0) GO TO 116
01072   299*          IL=((NBLOCK-28)/3)+1
01073   300*          TTOS=0                                                    *WS 2/71
01074   301*          DO 93 I=1,IL
01077   302*          J=26+(I-1)*3
01100   303*          NEED(I)=ICASE(J)
01101   304*          TOS(I)=(FLOAT(ICASE(J+1)))/240.
01102   305*          TTOS=TTOS+ICASE(J+1)                                      *WS 2/71
01103   306*          DELTA(I)=FLOAT(ICASE(J+2))/100.
01104   307*       93 CONTINUE
01106   308*          KL=IL
01107   309*          IF(ICASE(17).GT.0.AND.IL.GT.1) KL=IL-1
```

```
01111   310*        WRITE(5,595) (MEED(I),EOS(I),DELTA(I),I=1,IL)
01121   311*    595 FORMAT(5(I2,F5.4,F4.2))
01122   312*        IF(KL.LT.IL) WRITE(5,596) MEED(IL),EOS(IL),DELTA(IL)
01130   313*    596 FORMAT(10I5,110)
01131   314*    116 NCAS=NCAS+1
01132   315*        STATN=ICASE(2)                                          *WS 2/71
01133   316*        IF(STATN.LT.0)STATN=75                                  *WS 2/71
01135   317*        LSTSTN=MAX0(LSTSTN,STATN)                               *WS 2/71
01136   318*        VLDMTX(STATN,1)=VLDMTX(STATN,1)+1                       *WS 2/71
01137   319*        VLDMTX(STATN,2)=VLDMTX(STATN,2)+ICASE(15)               *WS 2/71
01140   320*        IF(ICASE(20).GT.0)VLDMTX(STATN,3)=VLDMTX(STATN,3)+1     *WS 2/71
01142   321*        IF(ICASE(21).GT.0)VLDMTX(STATN,4)=VLDMTX(STATN,4)+1     *WS 2/71
01144   322*        IF(ICASE(17).GT.0)VLDMTX(STATN,5)=VLDMTX(STATN,5)+1     *WS 2/71
01146   323*        VLDMTX(STATN,6)=VLDMTX(STATN,6)+TTOS                    *WS 2/71
01147   324*        VLDMTX(STATN,7)=VLDMTX(STATN,7)+ICASE(22)               *WS 2/71
01150   325*        NPRIOR=ICASE(14)+7                                      *WS 2/71
01151   326*        VLDMTX(STATN,NPRIOR)=VLDMTX(STATN,NPRIOR)+1             *WS 2/71
01152   327*        CAS2DA(STATN)=CAS2DA(STATN)+1                           *WS 2/71
01153   328*        NDAY=IDAY
01154   329*        IF(ABS(XLAMB).LE.1.E-5)GOTO76                           *WS 2/71
01156   330*        DEL=(LOG(RN)/-(XLAMB))*60.
01157   331*        IF(ISWIT(6).EQ.1) GO TO 115
01161   332*        IDELTA=DEL
01162   333*        DELTE=(3./XLAMB)*60.
01163   334*        IDELTE=DELTE
01164   335*        IF(IDELTA.GT.IDELTE)  DEL=DELTE
01166   336*     75 IDELTA=DEL
01167   337*        ZMIN=(ITIME/100)*60+AMOD(TIME,100.)
01170   338*        TMIN=ZMIN+DEL
01171   339*        TIME=(IFIX(TMIN)/60)*100+AMOD(TMIN,60.)
01172   340*        ITIME=TIME
01173   341*        GO TO 116
01174   342*     76 DEL=61.
01175   343*        IF(ISWIT(6).EQ.1) GO TO 115
01177   344*        GO TO 75
01200   345*    115 IDELTA=DEL
01201   346*        IF(IDELTA.LT.60) GO TO 75
01203   347*        DEL=60.
01204   348*        IDELTA=DEL
01205   349*        CALL ADTIME
01206   350*    118 CALL NTRAN(IUNIT,10)
01207   351*        IF(ITIME.GT.2400) GO TO 83
01211   352*        GO TO 82
01212   353*     83 JDAY=ITIME/2400
01213   354*        TIME=AMOD(TIME,2400.)
01214   355*        ITIME=TIME
01215   356*        IDAY=IDAY+JDAY
01216   357*        DO225STATN=1,NS                                         *WS 2/71
01221   358*        NC2D=0                                                  *WS 2/71
01222   359*    222 IF(NC2D.EQ.CAS2DA(STATN))GOTO224                        *WS 2/71
01224   360*        IF(NC2D.EQ.6)GOTO224                                    *WS 2/71
01226   361*        NC2D=NC2D+1                                             *WS 2/71
01227   362*        GOTO222                                                 *WS 2/71
01230   363*    224 CAS2DA(STATN)=0                                         *WS 2/71
01231   364*    225 VLDMTX(STATN,NC2D+13)=VLDMTX(STATN,NC2D+13)+1           *WS 2/71
01233   365*     82 IF(ISWIT(10).EQ.1) GO TO 91
01235   366*        IF(IDAY.GT.ISCED)  GO TO 90
01237   367*        IF(IDAY.EQ.ISCED.AND.ITIME.GT.ISCEH) GO TO 92
```

```
01241   365*          GO TO 38
01242   369*       92 IDAY=IDAY+1                                          *WS 3/71
01243   370*       91 IF(NCAS  .LT. MAXCA) GO TO 88
01245   371*       90 IF(ISWIT(3).EQ.1) GO TO 28
01247   372*          IDNO=2
01250   373*          IHOUR=0                                                *WS 3/71
01251   374*          IMIN=10                                                *WS 3/71
01252   375*          WRITE(8,580) IDNO,IDAY,IHOUR,IMIN
01260   376*          WRITE(6,580)  IDNO,IDAY,IHOUR,IMIN                     *WS 4/71
01266   377*          CALL NTRAN(8,9)                                        *WS 3/71
01267   378*       28 I=-1                                                   *WS 4/71
01270   379*      103 I=I+1
01271   380*          IF(I.GT.NDAY) GO TO 111
01273   381*          DO 112 J=1,NSPP
01276   382*          IF(I.GE.NSPPS(J).AND.I.LE.NSPPE(J)) GO TO 102
01300   383*      112 CONTINUE
01302   384*          J=5
01303   385*          GO TO 103
01304   386*      102 J=1
01305   387*      103 K=MOD(I,7)
01306   388*          IF(K.EQ.0.OR.K.EQ.6) J=J+2
01310   389*          IF(K.EQ.0.OR.K.EQ.6) GO TO 104
01312   390*          DO 105 K=1,NOHOLS
01315   391*          IF(I.EQ.IHOL(K)) GO TO 107
01317   392*      105 CONTINUE
01321   393*          GO TO 104
01322   394*      107 J=J+2
01323   395*      104 ZNDAYS(J)=ZNDAYS(J)+1
01324   396*          GO TO 106
01325   397*      111 DO 108 J=1,7,2
01330   398*          ZNDAYS(J+1)=ZNDAYS(J)
01331   399*      108 CONTINUE
01333   400*          WRITE(6,625)
01335   401*      625 FORMAT(//1X,'NUMBER OF SCENARIO DAYS IN EACH OF THE 8 BOXES'//)
01336   402*          WRITE(6,510) (ZNDAYS(J),J=1,8)
01344   403*          WRITE(6,630)
01346   404*      630 FORMAT(//1X,'HOURLY MEAN ARRIVAL RATE OF GENERATED CASES'//)
01347   405*          DO 109 I=1,8
01352   406*          DO 110 J=1,24
01355   407*          IF(ZNDAYS(I).LE.1.E-3)GOTO114                          *WS 2/71
01357   408*          XNLAMB(I,J)=NEWCAS(I,J)/ZNDAYS(I)
01360   409*          GOTO110                                                *WS 2/71
01361   410*      114 XNLAMB(I,J)=0.0                                        *WS 2/71
01362   411*      110 CONTINUE
01364   412*          WRITE(6,550) (XNLAMB(I,J),J=1,24)
01372   413*      109 CONTINUE
01374   414*      552 FORMAT(1X,I6)
01375   415*          WRITE(6,640)
01377   416*      640 FORMAT(//1X,'NUMBER OF CASES IN EACH BOX BY HOUR'//)
01400   417*          WRITE(6,553) ((NEWCAS(I,J),J=1,24),I=1,8)
01411   418*      553 FORMAT(1X,24I4)
01412   419*          IF(ISWIT(9).EQ.0) GO TO 101
01414   420*          IF(ISCEN.EQ.1)GOTO27                                   *WS 3/71
01416   421*          CALLNTRAN(8,8,-2)                                      *WS 3/71
01417   422*          CALLNTRAN(8,8,1)                                       *WS 3/71
01420   423*          GOTO100                                                *WS 3/71
01421   424*       27 CALLNTRAN(8,10)                                        *WS 3/71
01422   425*      100 READ(8,580)   IDNO,IDAY,IHOUR,IMIN
```

```
01430   426*          WRITE(6,561)  IDNO,IDAY,IHOUR,IMIN
01436   427*      561 FORMAT(1X,I3,I3,I3,I2)
01437   428*          INUMBA=0
01440   429*       95 INUMBA=INUMBA+1
01441   430*          READ (8,590) IDNO,IDAY,IHOUR,IMIN,(NCASE(I),I=1,12)
01453   431*          WRITE(6,590) IDNO,IDAY,IHOUR,IMIN,(NCASE(I),I=1,12)
01465   432*          READ (8,591) (NCASE(I),I=13,23)
01473   433*          WRITE(6,591) (NCASE(I),I=13,23)
01501   434*          JJ=NCASE(12)
01502   435*          IF(JJ.EQ.0) GO TO 98
01504   436*          K=JJ
01505   437*          IF(NCASE(14).GT.0.AND.K.GT.1) JJ=K-1
01507   438*          READ (8,595) (NEED(I),TOS(I),DELTA(I),I=1,JJ)
01517   439*          WRITE(6,595) (NEED(I),TOS(I),DELTA(I),I=1,JJ)
01527   440*          IF(JJ.EQ.K) GO TO 98
01531   441*          READ(8,595) NEED(K),TOS(K),DELTA(K)
01536   442*          WRITE(6,595) NEED(K),TOS(K),DELTA(K)
01543   443*      590 FORMAT(1X,I3,I4,I3,I2,I3,11I5)
01544   444*      591 FORMAT(1X,10I5,I10)
01545   445*      595 FORMAT(1X,6(I2,2F4.1))
01546   446*       98 IF(INUMBA.LT.   50)  GO TO 95
01550   447*          CALL NTRAN(8,8,1)                                        *WS 3/71
01551   448*      101 WRITE(6,635)
01553   449*      635 FORMAT(//1X,'FINAL RANDOM NUMBER (OCTAL)'//)
01554   450*          WRITE(6,592) B(1),B(2)
01560   451*      592 FORMAT(2(1X,O12))
01561   452*          WRITE(6,9001)                                            *WS 2/71
01563   453*     9001 FORMAT('1STA.  NO.OF  NO.OF  NO.W/  NO.W/   NO.   TOTAL  TOTAL  NO*WS 3/71
01563   454*         1.OF CASES W/ PRIORITY EQUAL TO:    NO. OF DAYS W/ N CASES OCCURRIN*WS 3/71
01563   455*         2G'/'  NO.  CASES  NEEDS  S1S>0  S2S>0  TOWED   TOS  ML.SRCH       *WS 3/71
01563   456*         3   1   2   3   4   5            0   1   2   3     4     5    >5'/)*WS 3/71
01564   457*          K=0                                                      *WS 2/71
01565   458*          DO900I=1,LSTSTN                                          *WS 2/71
01570   459*          IF(VLDMTX(I,1).EQ.0)GOTO900                              *WS 2/71
01572   460*          WRITE(6,9002)I,(VLDMTX(I,J),J=1,NV)                      *WS 2/71
01601   461*          K=K+1                                                    *WS 2/71
01602   462*          DO899J=1,NV                                              *WS 2/71
01605   463*      899 TOTAL(J)=TOTAL(J)+VLDMTX(I,J)                            *WS 2/71
01607   464*      900 CONTINUE                                                 *WS 2/71
01611   465*          DO 905J=15,19                                            *WS 2/71
01614   466*      905 TOTAL(J)=TOTAL(J)/K                                      *WS 2/71
01616   467*          WRITE(6,9003)TOTAL                                       *WS 2/71
01624   468*     9002 FORMAT(1X,I3,I7,I6,2I7,I8,2I7,9X,5I4,6X,I7,3I4,3I6)      *WS 2/71
01625   469*     9003 FORMAT(/' TOTAL',I5,I6,2I7,I8,2I7,9X,5I4,I13,3I4,4I6)    *WS 2/71
01626   470*          WRITE(6,9001)                                            *WS 2/71
01630   471*          DO260I=1,NA                                              *WS 2/71
01633   472*          L=0                                                      *WS 2/71
01634   473*          DO255J=1,NS                                              *WS 2/71
01637   474*          IF(AGRAGB(J).NE.I)GOTO255                                *WS 2/71
01641   475*          IF(VLDMTX(J,1).EQ.0)GOTO255                              *WS 2/71
01643   476*          L=L+1                                                    *WS 2/71
01644   477*          DO253K=1,NV                                              *WS 2/71
01647   478*      253 TEMP(K)=TEMP(K)+VLDMTX(J,K)                              *WS 2/71
01651   479*      255 CONTINUE                                                 *WS 2/71
01653   480*          IF(TEMP(1).EQ.0)GOTO260                                  *WS 2/71
01655   481*          DO256K=13,19                                             *WS 2/71
01660   482*      256 TEMP(K)=TEMP(K)/L                                        *WS 2/71
01662   483*          WRITE(6,9002)I,(TEMP(K),K=1,NV)                          *WS 2/71
```

```
01671   484*        DO258K=1,NV                                          *WS 2/71
01674   485*    258 TEMP(K)=0                                            *WS 2/71
01676   486*    260 CONTINUE                                             *WS 2/71
01700   487*   1000 WRITE(6,9003)TOTAL                                  *WS 2/71
01707   488*        CALLEXIT                                             *WS 2/71
01710   489*        END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   2 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   2 SEC.

TOTAL COMPILATION TIME =   6  SEC
```

```
BIT FOR BOX,BOX                                                                                    08 JUN 71          18:08:29.181
UNIVAC 1108 FORTRAN V LEVEL   2206 0013 F5013P
THIS COMPILATION WAS DONE ON 08 JUN 71 AT 18:08:29


   SUBROUTINE BOX        ENTRY POINT 000153

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000176
         0000    *DATA    000024
         0002    *BLANK   116563
         0003    BIX      002733


   EXTERNAL REFERENCES (BLOCK, NAME)

         0004    NWDU$
         0005    NIO2$
         0006    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000035 1176        0001   000070 2L          0001   000101 3L          0000   000067 4F          0003 I 000000 CAL
    0002 R 116161 CBOX        0002 R 116131 CPER        0000 I 000005 I           0002 I 000001 IADR        0002 I 116171 ICASE
    0000 I 000002 ID          0003 I 002725 IEWPP       0000 I 000001 IM          0003 I 002720 ISWPP       0000 I 000003 IT
    0000 I 000004 IWK         0000 I 000000 IY          0000 I 000006 J           0002 I 106241 NADR        0002 I 116101 NCASES
    0002 I 000000 NODRUM      0003 I 002732 NPP


00101      1*          SUBROUTINE BOX(IBOX,NT)
00103      2*          PARAMETER NP=24,NN=45,MXBX=4500,NXBX=1000,NH=20,NCP=250,NS=75      *WS 2/71
00103      3*        1          ,NV=18                                                    *WS 2/71
00104      4*          INTEGER CAL(12,31,4)
00105      5*          DIMENSION IADR(8,MXBX),NADR(4,NXBX),CPER(NP),CBOX(8),NCASES(NP)
00106      6*          DIMENSION ICASE(NCP)
00107      7*          COMMON/BIX/              CAL,ISWPP(5),IEWPP(5),NPP
00110      8*          COMMON NODRUM,IADR,NADR,NCASES,CPER,CBOX,ICASE
00111      9*          IY=MOD(ICASE(4),10)-5
00112     10*          IM=ICASE(4)/10
00113     11*          ID=ICASE(5)/10000
00114     12*          IT=MOD(ICASE(5),10000)
00115     13*          IWK=CAL(IM,ID,IY)
00116     14*          DO 1 I=1,NPP
00121     15*          IF(IWK.GE.ISWPP(I).AND.IWK.LE.IEWPP(I)) GO TO 2
00123     16*        1 CONTINUE
00125     17*          J=5
00126     18*          IF(MOD(IWK,2).NE.0) J=7
00130     19*          GO TO 3
00131     20*        2 J=1
00132     21*          IF(MOD(IWK,2).NE.0) J=3
00134     22*        3 IF(IT.LT.0800.OR.IT.GE.2000) J=J+1
00136     23*          NT=(IT/100)+1
```

```
00137    24*          IBOX=0
00140    25*          IF(IBOX.GT.8)  WRITE(6,4) IBOX,IY,IM,ID,IT,NT,IWK
00152    26*        4 FORMAT(1X,7I6)
00153    27*          RETURN
00154    28*          END
```

END OF UNIVAC 1108 FORTRAN V COMPILATION. 0 *DIAGNOSTIC* MESSAGE(S)

```
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   0 SEC.

TOTAL COMPILATION TIME =   0  SEC
```

```
WIT FOR ADD,ADD                                                                    08 JUN 71          18:08:30.119
UNIVAC 1108 FORTRAN V LEVEL   2L05 0018 F50149
THIS COMPILATION WAS DONE ON 08 JUN 71 AT 18:08:30


   SUBROUTINE ADD        ENTRY POINT 000156

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001    *CODE    000164
         0000    *DATA    000042
         0002    *BLANK   116563


   EXTERNAL REFERENCES (BLOCK, NAME)

         0003    NTRAN
         0004    RAND
         0005    BOX
         0006    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000004 107G        0001   000034 117G        0000 R 000007 B          0002 R 116161 CBOX       0002 R 116131 CPER
    0000 R 000002 CPERC       0000 I 000000 I           0002 I 000001 IADR       0000 I 000001 IAUNIT     0000 I 000014 IBOX
    0002 I 116171 ICASE       0000 I 000016 IUNIT       0000 I 000005 J          0000 I 000017 K          0000 I 000013 L
    0002 I 106241 NADR        0000 I 000011 NBLOCK      0000 I 000010 NCANO      0000 I 000003 NCASE      0002 I 116101 NCASES
    0002 I 000000 NODRUM      0000 I 000012 NSPACE      0000 I 000015 NT         0000 I 000004 NUM        0000 I 000020 NUPPER
    0004 R 000000 RAND        0000 R 000006 RN


00101     1*          SUBROUTINE ADD
00103     2*          PARAMETER NP=24,NN=45,MXBX=4500,NXBX=1000,NH=20,NCP=250,NS=75      *WS 2/71
00103     3*         1          ,NV=18                                                  *WS 2/71
00104     4*          DIMENSION IADR(8,MXBX),NADR(4,NXBX),CPER(NP),CBOX(8),NCASES(NP)
00104     5*         1,ICASE(NCP)
00105     6*          COMMON NODRUM,IADR,NADR,NCASES,CPER,CBOX,ICASE
00106     7*          DO 1 I=1,NODRUM
00111     8*          IAUNIT=I+17
00112     9*          CALL NTRAN(IAUNIT,10)
00113    10*          CPERC=CPER(I)
00114    11*          NCASE=NCASES(I)
00115    12*          NUM=NCASE*CPERC
00116    13*          DO 2 J=1,NUM
00121    14*          RN=RAND(0,3)
00122    15*          NCANO=RN*NCASE+1
00123    16*          NBLOCK=NADR(IAUNIT,NCANO+1)-NADR(IAUNIT,NCANO)
00124    17*          NSPACE=NADR(IAUNIT,NCANO)-1
00125    18*          CALL NTRAN(IAUNIT,6,NSPACE)
00126    19*          CALL NTRAN(IAUNIT,2,NBLOCK,ICASE(1),L)
00127    20*          CALL BOX(IBOX,NT)
00130    21*          IUNIT=IBOX+9
00131    22*          CBOX(IBOX)=CBOX(IBOX)+1
```

```
00132    23*          K=CBOX(IBOX)
00133    24*          IADR(IBOX,K+1)=IADR(IBOX,K)+NBLOCK
00134    25*          CALL NTRAN(IUNIT,1,NUPPER,ICASE(1),L)
00135    26*          CALL NTRAN(IAUNIT,10)
00136    27*        2 CONTINUE
00140    28*        1 CONTINUE
00142    29*          RETURN
00143    30*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   0 SEC.


TOTAL COMPILATION TIME =   0  SEC
```

```
SUBROUTINE ADTIME     ENTRY P INT 0004 2

STORAGE USED (BLOCK, NAME, LENGTH)

      0001   *CODE   000407
      0000   *DATA   000043
      0002   *BLANK  000000
      0003   AIX     000367


EXTERNAL REFERENCES (BLOCK, NAME)

      0004   RAND
      0005   ALOG
      0006   NERR3$


STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000007 1L          0001    000303 10L         0001    000121 131G        0001    000165 144G        0001    000107 2L
 0001    000175 7L          0001    000177 8L          0001    000314 9L          0003 R 000360 B           0003 R 000366 DEL
 0000 I 000014 I30X         0003 I 000364 IDAY         0003 I 000020 IDELTA       0003 I 000021 IHOL        0003 I 000045 ILIST
 0000 I 000002 IOLDTM       0003 I 000356 ISCED        0003 I 000357 ISCEH        0003 I 000000 ISWIT       0003 I 000355 ITIME
 0000 I 000007 I1           0000 I 000011 I2           0000 I 000013 I3           0000 I 000010 J           0000 I 000006 JDAY
 0000 I 000015 JTIME        0000 I 000012 K            0000 I 000000 KDAY         0003 R 000055 LAMBDA      0000 I 000003 N
 0000 R 000016 RN           0003 R 000365 TIME         0000 R 000005 TMIN         0000 R 000017 XLAMB       0000 R 000004 ZMIN
 0000 R 000001 ZTIME
```

```
00101     1*          SUBROUTINE ADTIME
00103     2*          PARAMETER NP=24,NN=45,MXBX=4500,NXBX=1000,NH=20,NCP=250,NS=75      *WS 2/71
00103     3*         1          ,NV=18                                                  *WS 2/71
00104     4*          REAL LAMBDA(8,NP)
00105     5*          DIMENSION ISWIT(10),NSPPS(3),NSPPE(3),IHOL(NH),ILIST(2,2,2),B(2)
00106     6*          COMMON/AIX/ ISWIT,NSPPS,NSPPE,IDELTA,IHOL,ILIST,LAMBDA,ITIME,
00106     7*         1ISCED,ISCEH,B,NSPP,NOHOLS,IDAY,TIME,DEL
00107     8*          KDAY=IDAY
00110     9*          ZTIME=TIME
00111    10*          IOLDTM=ITIME
00112    11*          N=0
00113    12*        1 N=N+1
00114    13*          ZMIN=(ITIME/100)*60+AMOD(TIME,100.)
00115    14*          TMIN=ZMIN+DEL
00116    15*          TIME=(IFIX(TMIN)/60)*100+AMOD(TMIN,60.)
00117    16*          ITIME=TIME
00120    17*          IF(ITIME.LE.2400)  GO TO 2
00122    18*          JDAY=ITIME/2400
00123    19*          TIME=AMOD(TIME,2400.)
```

```
00124    20*          ITIME=TIME
00125    21*          KDAY=KDAY+JDAY
00126    22*        2 I1=2
00127    23*          I1=2
00130    24*          DO 5 J=1,NSPP
00133    25*          IF(KDAY.GE.NSPPS(J).AND.KDAY.LE.NSPPE(J)) I1=1
00135    26*        5 CONTINUE
00137    27*          I2=1
00140    28*          K=MOD(KDAY,7)
00141    29*          IF(K.EQ.0.OR.K.EQ.6) I2=2
00143    30*          DO 6 J=1,NOHOLS
00146    31*          IF(KDAY.EQ.IHOL(J)) GO TO 7
00150    32*        6 CONTINUE
00152    33*          GO TO 8
00153    34*        7 I2=2
00154    35*        8 I3=1
00155    36*          IF(ITIME.LT.0800.OR.ITIME.GE.2000) I3=2
00157    37*          IBOX=ILIST(I1,I2,I3)
00160    38*          JTIME=(ITIME/100)+1
00161    39*          RN=RAND(0,8)
00162    40*          XLAMB=LAMBDA(IBOX,JTIME)
00163  *DIAGNOSTIC*  THE TEST FOR EQUALITY BETWEEN NON-INTEGERS MAY NOT BE MEANINGFUL.
00163    41*          IF(XLAMB.EQ.0.) GO TO 10
00165    42*          DEL=(LOG(RN)/-XLAMB)*60.
00166    43*          IDELTA=DEL
00167    44*          IF(IDELTA.LT.60) GO TO 9
00171    45*          DEL=60.
00172    46*          IDELTA=DEL
00173    47*          GO TO 1
00174    48*       10 DEL=61.
00175    49*          IDELTA=DEL
00176    50*          GO TO 1
00177    51*        9 ZTIME=(IOLDTM/100)*60+AMOD(ZTIME,100.)
00200    52*          TMIN=ZTIME+N*60+DEL
00201    53*          TIME=(IFIX(TMIN)/60)*100+AMOD(TMIN,60.)
00202    54*          ITIME=TIME
00203    55*          RETURN
00204    56*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     1 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.


TOTAL COMPILATION TIME =   1  SEC
```

```
ASSEMBLED BY NIVAC 1[illegible] EX[illegible] AT ASS[illegible]  [illegible]
THIS ASSEMBLY [illegible]
000001      00    000000  71 13  1 13 1  [illegible]      AUD*  DL     13,*1,11
000002            000001  71 12  1 [illegible]                  DS     13,I[illegible]
000003            000002  27 17 14 00 1 [illegible]              L,017  12,[illegible]
000004            000003  27 00 15 00 0 [illegible]              L      13,I5+1
000005            000004  32 00 [illegible]                      MF     13,
000006            000005  73 00 12 00 0 000001                   SSC    14,1
000007            000006  27 00 17 00 0 000000                   L      15,I5
000008            000007  30 00 [illegible] 00 0 000032          MI     15,[illegible]
000009            000010  71 10 00 00 0 000017                   DA     12,15
000010            000011  73 12 01 00 0 000001                   LSSL   13,1
000011            000012  73 02 01 00 0 000001                   SSL    13,1
000012            000013  71 12 01 00 0 000000                   DS     13,I5
000013            000014  27 17 14 00 0 000170                   L,017  12,0170
000014            000015  76 05 00 00 0 000015                   LCF    12,13
000015            000016  27 00 14 00 0 000015                   L      12,13
000016            000017  71 13 01 00 0 000000                   DL     13,I5
000017            000020  71 12 01 13 1 000001                   DS     13,*1,11
000018            000021  51 00 [illegible] 13 1 [illegible]     TMZ    *0,11
000019            000022  74 04 [illegible] 13 0 000003          J      3,11
000020            000023  24 09 14 00 0 000015                   A      12,9II
000021            000024  76 01 00 00 0 000004                   FAM    12,0IF
000022            000025  74 04 00 13 0 000003                   J      3,11
000023      01    000000  154447730601            X(1),I5   +015444773060 1
000024            000001  255751305264                      +025575130526 4
000025            000002  011060471625            Y         +011060471625
000026            000003  001000000000            BIT       +001000000000
000027            000004  200777777777            ONE       +020077777777 7
000028                                                      END
```

```
WIT FOR SELECT,SELECT                                                                    08 JUN 71          18:08:33.120
UNIVAC 1108 FORTRAN V LEVEL   2206 0018 F50138
THIS COMPILATION WAS DONE ON 08 JUN 71 AT 18:08:33


   SUBROUTINE SELECT    ENTRY POINT 000073

   STORAGE USED (BLOCK, NAME, LENGTH)

         0001   *CODE   000076
         0000   *DATA   000026
         0002   *BLANK  116563
         0003   SEAL    000001


   EXTERNAL REFERENCES (BLOCK, NAME)

         0004   NTRAN
         0005   NWDU$
         0006   NIO2$
         0007   NSTOP$
         0010   NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000051 1L          0001   000045 20L         0001   000054 25L         0000   000004 507F        0002 R 116161 CBOX
    0002 R 116131 CPER        0002 I 000001 IADR        0000 I 000001 IAUNIT      0002 I 116171 ICASE       0000 I 000000 IDRUM
    0000 I 000002 K           0000 I 000003 L           0002 I 106241 NADR        0002 I 116101 NCASES      0002 I 000000 NODRUM
    0003 I 000000 NUPPER


00101     1*          SUBROUTINE SELECT
00103     2*          PARAMETER NP=24,NN=45,MXBX=4500,NXBX=1000,NH=20,NCP=250,NS=75      *WS 2/71
00103     3*         1          ,NV=18                                                   *WS 2/71
00104     4*          COMMON/SEAL/ NUPPER
00105     5*          DIMENSION IADR(8,MXBX),NADR(4,NXBX),CPER(NP),CBOX(8),NCASES(NP)
00105     6*         1,ICASE(NCP)
00106     7*          COMMON NODRUM,IADR,NADR,NCASES,CPER,CBOX,ICASE
00106     8*    C
00106     9*    C  THESE ARE USER FURNISHED STATEMENTS
00106    10*    C
00107    11*          IDRUM=0
00110    12*          IF(ICASE(14).GT.3)  IDRUM=1
00112    13*          IF(ICASE(17).GT.0)  IDRUM=2
00114    14*          IF(IDRUM.EQ.0) GO TO 1
00116    15*          IAUNIT=17+IDRUM
00117    16*          NCASES(IDRUM)=NCASES(IDRUM)+1
00120    17*          K=NCASES(IDRUM)+1
00121    18*          NADR(IDRUM,K)=NADR(IDRUM,K-1)+NUPPER
00122    19*          CALL NTRAN(IAUNIT,1,NUPPER,ICASE(1),L)
00123    20*       20 IF(L+1) 25,20
00126    21*        1 RETURN
00127    22*       25 WRITE(6,507)
```

```
00131    23*      507 FORMAT (2X,'ERROR IN TRANSMISSION IN SELECT')
00132    24*          STOP
00133    25*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   0 SEC.


TOTAL COMPILATION TIME =   0  SEC
```

HIST

Certain information pertinent to the specific simulation being carried out must be indicated in the first two source cards of HIST. In card 1, NMO is set equal to the number of months the simulation will be exercised. In card 2, IMO is set to the number of the month the simulation begins (January = 1, etc.), IYR is the last digit of the year it begins and ISTANO is the station number of the C-130 station which would respond to any cases involving C-130's in the simulation.

```
MAIN PROGRAM

STORAGE USED (BLOCK, NAME, LENGTH)

      0001    *CODE    000773
      0000    *DATA    001104
      0002    *BLANK   000000
      0003    CONV     001757


EXTERNAL REFERENCES (BLOCK, NAME)

      0004    NTRAN
      0005    CONVER
      0006    NWD1$
      0007    NIO2$
      0010    NRBD$
      0011    NIO1$
      0012    NSTOP$


STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0001    000267 1L        0001    000336 100L      0001    000375 110L      0001    000012 116G      0001    000503 120L
 0001    000534 122L      0001    000574 125L      0001    000662 130L      0001    000036 137G      0001    000771 140L
 0001    000326 15L       0001    000100 160G      0001    000121 171G      0001    000072 2L        0001    000263 233G
 0001    000356 264G      0001    000371 273G      0001    000302 3L        0001    000412 304G      0001    000420 312G
 0001    000507 335G      0001    000526 346G      0001    000556 362G      0001    000322 4L        0001    000704 427G
 0001    000723 436G      0001    000143 44L       0001    000163 45L       0001    000444 5L        0001    000220 50L
 0001    000762 500L      0001    000625 6L        0001    000240 60L       0000    001036 600F      0000    001046 601F
 0000    001053 602F      0000    001032 603F      0000    001045 605F      0000    001030 606F      0000    001034 608F
 0001    000656 7L        0000    001051 701F      0000 I  001004 I         0000 I  001013 IACEPT    0003 I  000000 ICASE
 0000 I  001022 IDAY      0000 I  001011 IDNO      0000 I  000150 IEND      0000 I  001010 IFLAG     0003 I  001756 IKAP
 0000 I  001025 ILOW      0000 I  000000 IMO       0000 I  001014 IMONTH    0000 I  001027 INDEX     0000 I  001007 IOPFAC
 0000 I  001012 IPOINT    0000 I  001026 IS        0000 I  000003 ISTANO    0000 I  000004 ISTART    0000 I  001017 ISTAT
 0000 I  000014 ISWIT     0000 I  001023 ITIME     0000 I  001015 IYEAR     0000 I  000002 IYR       0000 I  000001 I$
 0003 I  001751 I1        0003 I  001752 I2        0000 I  001005 I2END     0003 I  001753 I3        0003 I  001754 I4
 0000 I  001006 J         0000 I  000770 JMO       0000 I  001024 JNO       0000 I  001021 K         0003 I  001755 KAP
 0000 I  001020 NBLOCK    0003 I  000764 NCASE     0000 I  000624 NCOUNT    0000 I  001016 NEXTRA    0000 I  000460 NTIME
 0003 I  001750 NUPPER
```

```
00101     1*          PARAMETER NMO=1
00103     2*          DIMENSION IMO(NMO)/8/,IYR(1)/8/,ISTANO(1)/81/
00107     3*          DIMENSION ICASE(500),NCASE(500),ISTART(100),IEND(100),ISWIT(100),
00107     4*         1NTIME(100),NCOUNT(100),JMO(12)
00110     5*          COMMON/CONV/ ICASE,NCASE,NUPPER,I1,I2,I3,I4,KAP,IKAP
00111     6*          DATA JMO/31,28,31,30,31,30,31,31,30,31,30,31/
00113     7*          KAP=0
```

```
00114      8*          IKAP=0
00115      9*          DO 1 I=1,100
00120     10*          NCOUNT(I)=0
00121     11*        1 ISWIT(I)=0
00123     12*          I1=3
00124     13*          I2=0
00125     14*          I3=0
00126     15*          I4=0
00127     16*          WRITE(8,601) I1,I2,I3,I4
00135     17*          I2END=0
00136     18*          DO 11 I=1,NMO
00141     19*          J=IMO(I)
00142     20*          I2END=I2END+JMO(J)
00143     21*          IF (IMO(I).EQ.2.AND.IYR(1).EQ.8) I2END=I2END+1
00145     22*       11 CONTINUE
00147     23*          I1=1
00150     24*          CALL NTRAN(35,10)
00151     25*          IOPFAC=0
00152     26*          IFLAG=0
00153     27*          IDNO=1
00154     28*          IPOINT=1
00155     29*        2 IACEPT=0
00156     30*          READ (7) (ICASE(I),I=1,28)
00164     31*          IF(ICASE(1).EQ.99) GO TO 100
00166     32*          IMONTH=ICASE(4)/10
00167     33*          IYEAR=ICASE(4)-IMONTH*10
00170     34*          DO 43 I=1,NMO
00173     35*          IF (IMONTH.EQ.IMO(I).AND.IYEAR.EQ.IYR(1)) GO TO 44
00175     36*       43 CONTINUE
00177     37*          IACEPT=1
00200     38*          GO TO 60
00201     39*       44 IF(IFLAG.EQ.2) GO TO 60
00203     40*          IF(IFLAG.EQ.1) GO TO 50
00205     41*          IF(ICASE(3)/10000.NE.2) IFLAG=1
00207     42*       45 IF(IOPFAC.EQ.ICASE(2).AND.IFLAG.EQ.0) GO TO 60
00211     43*          IOPFAC=ICASE(2)
00212     44*          IF(IDNO.NE.1) IEND(IDNO-1)=IPOINT-1
00214     45*          ISTART(IDNO)=IPOINT
00215     46*          IDNO=IDNO+1
00216     47*          GO TO 60
00217     48*       50 IF (ICASE(2).EQ.0.OR.ICASE(2).EQ.ISTANO(1)) IFLAG=2
00221     49*          IF (IFLAG.EQ.2) GO TO 45
00223     50*       60 NEXTRA=(ICASE(15)-1)*3
00224     51*          IF(ICASE(15).EQ.0) NEXTRA=0
00226     52*          NUPPER=29+NEXTRA-1
00227     53*          IF(NEXTRA.EQ.0) GO TO 10
00231     54*          READ(7) (ICASE(I),I=29,NUPPER)
00237     55*       10 IF(IACEPT.EQ.1)  GO TO 2
00241     56*          CALL NTRAN(35,1,28,ICASE(1),ISTAT)
00242     57*        3 IF(ISTAT+1) 500,3,
00245     58*          IF(NEXTRA.EQ.0) GO TO 15
00247     59*          NBLOCK=NUPPER-28
00250     60*          CALL NTRAN(35,1,NBLOCK,ICASE(29),ISTAT)
00251     61*        4 IF(ISTAT+1) 500,4,
00254     62*       15 IPOINT=IPOINT+NUPPER
00255     63*          NCOUNT(IDNO-1)=NCOUNT(IDNO-1)+1
00256     64*          GO TO 2
00257     65*      100 IDNO=IDNO-1
```

```
00260     68*          IEND(IDNO)=IPOINT-1
00261     67*          WRITE(6,603) I NO,(ISTART(I),IEND(I),I=1,IDNO)
00271     68*          WRITE(6,606) (NCOUNT(I),I=1,IDNO)
00277     69*      603 FORMAT(1X,39I3)
00300     70*      606 FORMAT(I4,(16I7))
00301     71*      110 CALL NTRAN(35,10)
00302     72*          WRITE(6,608) (ISWIT(I),I=1,IDNO)
00310     73*      608 FORMAT(T10,39I2)
00311     74*          DO 120 I=1,IDNO
00314     75*          IF(ISWIT(I).EQ.1) GO TO 120
00316     76*          K=ISTART(I)-1
00317     77*          CALL NTRAN (35,6,K)
00320     78*          CALL NTRAN(35,2,28,ICASE(1),ISTAT)
00321     79*        5 IF(ISTAT+1) 500,5,
00324     80*          IMONTH=ICASE(4)/10
00325     81*          IYEAR=ICASE(4)-IMONTH*10
00326     82*          IDAY=ICASE(5)/10000
00327     83*          ITIME=ICASE(5)-IDAY*10000
00330     84*          NTIME(I)=IYEAR*10**8+IMONTH*10**6+IDAY*10**4+ITIME
00331     85*          CALL NTRAN(35,10)
00332     86*      120 CONTINUE
00334     87*          DO 121 I=1,IDNO
00337     88*          JNO=I
00340     89*          IF(ISWIT(I).NE.1) GO TO 122
00342     90*      121 CONTINUE
00344     91*          WRITE(6,600) (ISWIT(I),I=1,IDNO)
00352     92*      600 FORMAT(T10,'ERROR IN ISWIT ARRAY',/(20I4))
00353     93*          STOP
00354     94*      122 ILOW=NTIME(JNO)
00355     95*          IS=JNO+1
00356     96*          IF(IS.GT.IDNO) IS=IDNO
00360     97*          INDEX=JNO
00361     98*          DO 125 I=IS,IDNO
00364     99*          IF(ISWIT(I).EQ.1) GO TO 125
00366    100*          IF(NTIME(I).GE.ILOW) GO TO 125
00370    101*          ILOW=NTIME(I)
00371    102*          INDEX=I
00372    103*      125 CONTINUE
00374    104*          WRITE(6,605) INDEX ,ILOW
00400    105*      605 FORMAT (2I10)
00401    106*          K=ISTART(INDEX)-1
00402    107*          CALL NTRAN(35,6,K)
00403    108*          CALL NTRAN(35,2,28,ICASE(1),ISTAT)
00404    109*        6 IF(ISTAT+1) 500,6,
00407    110*          NEXTRA=(ICASE(15)-1)*3
00410    111*          IF(ICASE(15).EQ.0) NEXTRA=0
00412    112*          NUPPER=29+NEXTRA-1
00413    113*          IF(NEXTRA.EQ.0) GO TO 130
00415    114*          NBLOCK=NUPPER-28
00416    115*          CALL NTRAN(35,2,NBLOCK,ICASE(29),ISTAT)
00417    116*        7 IF(ISTAT+1) 500,7,
00422    117*      130 CALL CONVER
00423    118*          ISTART(INDEX)=ISTART(INDEX)+NUPPER
00424    119*          NCOUNT(INDEX)=NCOUNT(INDEX)-1
00425    120*          WRITE(6,606) (NCOUNT(I),I=1,IDNO)
00433    121*          IF(ISTART(INDEX).GE.IEND(INDEX)) ISWIT(INDEX)=1
00435    122*          DO 135 I=1,IDNO
00440    123*          IF(ISWIT(I).EQ.0) GO TO 110
```

```
00442   124*      135 CONTINUE
00444   125*          I1=2
00445   126*          I2=I2E(1)
00446   127*          I3=0
00447   128*          I4=5
00450   129*          WRITE(3,601) I1,I2,I3,I4
00456   130*      601 FORMAT(I3,I4,I7,I2)
00457   131*          WRITE(6,701) I1,I2,I3,I4
00465   132*      701 FORMAT (1H0,4I3)
00466   133*          GO TO 140
00467   134*      500 WRITE(6,602)
00471   135*      602 FORMAT(T10,'ERROR IN NTRAN TRANSMISSSION')
00472   136*          STOP
00473   137*      140 END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   1 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   0 SEC.
PHASE 6 TIME =   1 SEC.

TOTAL COMPILATION TIME =   2  SEC
```

```
UNIVAC 1106 FORTRAN V LEVEL   2206 0018 F50189
THIS COMPILATION WAS DONE ON 07 JUN 71 AT 19:12:10


  SUBROUTINE CONVER     ENTRY POINT 000340

  STORAGE USED (BLOCK, NAME, LENGTH)

        0001   *CODE   000351
        0000   *DATA   000542
        0002   *BLANK  000000
        0003   CONV    001757


  EXTERNAL REFERENCES (BLOCK, NAME)

        0004   NWDU$
        0005   NIO1$
        0006   NIO2$
        0007   NERR3$


  STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

   0001   000002 166G      0001   000011 115G      0001   000016 123G      0001   000141 162G      0001   000153 171G
   0001   000172 252G      0001   000250 217G      0001   000315 244G      0001   000300 5L        0000   000474 580F
   0000   000463 585F      0000   000467 590F      0000   000471 595F      0001   000275 6L        0000   000506 600F
   0001   000307 7L        0000 R 000310 DELTA     0000 I 000454 I         0003 I 000000 ICASE     0003 I 001756 IKAP
   0000 I 000460 IL        0000 I 000455 IPREV     0003 I 001751 I1        0003 I 001752 I2        0000 I 000456 I2PREV
   0003 I 001753 I3        0003 I 001754 I4        0000 I 000457 I6        0000 I 000461 J         0003 I 001755 KAP
   0000 I 000462 KL        0003 I 000764 NCASE     0000 I 000000 NEED      0003 I 001750 NUPPER    0000 R 000144 TOS


00101     1*          SUBROUTINE CONVER
00103     2*          COMMON/CONV/ ICASE,NCASE,NUPPER,I1,I2,I3,I4,KAP,IKAP
00104     3*          DIMENSION ICASE(500),NCASE(500),NEED(100),TOS(100),DELTA(100)
00105     4*          DO 1 I=1,3
00110     5*          NCASE(I)=ICASE(I)
00111     6*        1 CONTINUE
00113     7*          NCASE(4)=ICASE(6)
00114     8*          DO 2 I=5,15
00117     9*          NCASE(I)=ICASE(I+3)
00120    10*        2 CONTINUE
00122    11*          DO 3 I=16,21
00125    12*          NCASE(I)=ICASE(I+4)
00126    13*        3 CONTINUE
00130    14*          NCASE(22)=0
00131    15*          NCASE(23)=ICASE(7)
00132    16*          I2=ICASE(5)/10000
00133    17*          IF(I2.LT.1.OR.I2.GT.31) GO TO 5
00135    18*          IF(ICASE(4).EQ.IPREV.AND.I2.LT.I2PREV) GO TO 7
00137    19*          IF(I2.LT.KAP) IKAP=IKAP+KAP
00141    20*          KAP=I2
```

```
00142    21*          I2=I2+IKAP-1
00143    22*          I2PREV=I2
00144    23*          IPREV=ICASE(4)
00145    24*          I6=ICASE(5)-(ICASE(5)/10000)*10000
00146    25*          I3=I6/100
00147    26*          I4=I6-I3*100
00150    27*          IF(I3.GT.23) GO TO 5
00152    28*          IF(I4.GT.59) GO TO 5
00154    29*          WRITE(8,585) I1,I2,I3,I4,(NCASE(I),I=1,12)
00166    30*      585 FORMAT(I3,I4,I3,I2,I3,11I5)
00167    31*          WRITE(8,590) (NCASE(I),I=13,23)
00175    32*      590 FORMAT(10I5,I10)
00176    33*          IF(ICASE(15).EQ.0)  GO TO 6
00200    34*          IL=((NUPPER-28)/3)+1
00201    35*          DO 4 I=1,IL
00204    36*          J=26+(I-1)*3
00205    37*          NEED(I)=ICASE(J)
00206    38*          TOS(I)=(FLOAT(ICASE(J+1)))/240.
00207    39*          DELTA(I)=FLOAT(ICASE(J+2))/100.
00210    40*        4 CONTINUE
00212    41*          KL=IL
00213    42*          IF(ICASE(17).GT.0.AND.IL.GT.1) KL=IL-1
00215    43*          WRITE(8,595) (NEED(I),TOS(I),DELTA(I),I=1,KL)
00225    44*      595 FORMAT(6(I2,F6.4,F4.2))
00226    45*          IF(KL.LT.IL) WRITE(8,595) NEED(IL),TOS(IL),DELTA(IL)
00234    46*        6 RETURN
00235    47*        5 WRITE(6,580) ICASE(5)
00240    48*      580 FORMAT(T10,'ERROR IN DATA  CASE REJECTED-DAY AND TIME =',I8)
00241    49*          GO TO 6
00242    50*        7 WRITE(6,600) (ICASE(I),I=1,28)
00250    51*      600 FORMAT(1X,14I9/)
00251    52*          GO TO 6
00252    53*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =   0 SEC.
PHASE 2 TIME =   0 SEC.
PHASE 3 TIME =   0 SEC.
PHASE 4 TIME =   0 SEC.
PHASE 5 TIME =   1 SEC.
PHASE 6 TIME =   0 SEC.


TOTAL COMPILATION TIME =   1  SEC
```

```
000001          +T CASE 8882      T ITOL 012/4 I  1RES   E /        RQUED *
000002          +                 T MMM  021/4 I  2COST  1 /2 F     EXCSO *
000003          +                 T NNN  022/4 I  3EIAT  1 /2 F     LISTO *
000004          +                 T XC   031/2*F  4TVEC  1 /2 F     NSET1 *
000005          +                 T YC   281/2*F  5IFLT  1 /2 I     SRHS1 *
000006          +                 T POB  041/3 I  6IB    1 /4 I     CQUE    *PRI    h
000007          +                 T UTYPE042/3 I  7PRIOR 1 /4 I     CSET    *COST   L
000008          +                 T L    043/3 I  8NCASE 1 /2 I     TSET    *DVEC   L
000009          +                 T S1S  051/4*I  9PR    1 /2 F     PSET    *PR     H
000010          +                 T S2S  052/4*I 10UTIL  1 /2 F
000011          +                 T STATN053/4 I 11TUTIL 1 /2 F
000012          +                 T AIR  054/4 I 12SRQUE 1 /2 I
000013          +                 T OST  061/2 F 13PTSET 1 /2 I
000014          +                 T RESA 063/4 I 14STSET 1 /2 I
000015          +                 T SWELL064/4 I 15PCSET 1 /2 I
000016          +                 T NEED 071/4 I 16SCSET 1 /2 I
000017          +                 T PRI  072/4 I 17PPSET 1 /2 I
000018          +                 T FLG  073/4 I 18SPSET 1 /2 I
000019          +                 T SIGNL074/4 I 19XR    1 /2*F
000020          +                 T PCQUE081/2 I 20YR    1 /2*F
000021          +                 T SCQUE082/2 I 21STN   1 /4 I
000022          +                 T FPRI 111/4 I 22TYPE  1 /4 I
000023          +                 T REA  112/4 I127DVEC  1 /2 F
000024          +                 T IDLOC113/4 I            /
000025          +                 T WIND 114/4 I            /
000026          +                 T NOINT121/4 I            /
000027          +                 T LOC  122/4 I            /
000028          +                 T COUNT123/4 I 23RST   E /
000029          +                 T IWAIT124/4 I 24AUT   1 /2 F
000030          +                 T VIS  131/4 I 25END   1 /2 F*
000031          +                 T IS2  132/4*I 26COSTD 1  2 F*
000032          +                 T OCCUR27    F 27RCOST 1 /4 I*
000033          +                 T NOCAS141/2 I 28SOA1  1 /2 F*
000034          +                 T OFSHR142/2 F 29SOA2  1 /2 F*
000035          +                 T FNSET151/2 I 30SOA3  1 /2 F*
000036          +                 T LNSET152/2 I 31SLIM  1 /4 I*
000037          +                 T FSRHS161/2 I 32TF    1 /2 F*
000038          +                 T LSRHS162/2 I 33MR    1 /2 F*
000039          +                 T GAMMA171/2 F120VOP   1  4 I*
000040          +                 T SEXCS172/2 I128SQTAG 1 /4 I*
000041          +                 T XCX  181/2*F129DLAY  1  2 F*
000042          +                 T YCY  182/2*F
000043          +                 T TINQ 25    F            /
000044          +                 T TQUE 132/2 F 34STA   E /
000045          +                 T TSM  221/2 F 35ACS   2 /4 I*
000046          +                 T TQUE1222/2 F 36ADJS  2 /4 I*
000047          +                 T COSTC231/2 F 37CUT   2 /4 I*
000048          +                 T TWAIT232/2 F 38XS    1 /2*F*
000049          +                 T TINT 282/2 F 39YS    1 /2*F*
000050          +                 T STINQ24    F 40AVGTW 1 /2 F
000051          +                 T TSVC 212/2 F 41BCREW 1 /4 I
000052          +                 T NQUE 211/4 I 42FAIL1 1 /4 I
000053          +                 T CNRES212/4 I 43FAIL2 1 /4 I
000054          +                 T VALUE26    I 44FAIL3 1 /2 I
```

```
000055          +                T ITOA 311/3 I 45NCAS  1 /4 I
000056          +                T BOX  312/3 I 46NEEDS 1 /2 I
000057          +                T SYTAG011/4 I 47MINTR 1  4 I
000058          +                T OPFAC313/3 I 48NSTBY 1 /4 I
000059          +                T DMERT32    F 49REST  2 /4 I*
000060          +N NOTIF8        N COMP 031/4 I 50SHIFT 2 /4 I*
000061          +                N NUMBR032/4 I 51UNPRO 1 /4 I
000062          +                N CAS  041/2 I 52USE   1 /2 F
000063          +                N SNSET042/2 I 54USHF  2 /2 F
000064          +                N XHAND051/2*F131DMRT  1  2 F
000065          +                N YHAND052/2*F132GRP   1  4 I
000066          +                N KRES 064/4 I133VCTR  1 /2 F
000067          +                N DELTA032/2 F134TNTOL 1 /2*F
000068          +                          /    53NEADS E /
000069          +                          /    55AVUS  1 /2 F
000070          +                          /    56TME   1 /2 F
000071          +                          /    57TOTME 1 /2 F
000072          +N NOTE 8        N TS   031/2 F           /
000073          +                N SSRHS032/2 I           /
000074          +                N SM   041/2 F           /
000075          +                N ESAC 042/2 I 58TOLER E /
000076          +                N SDAY 051/4 I 59TOL   1 /2 F*
000077          +                N SFLAG052/4 I 60TOLS  1 /2 F*
000078          +                N RSRC 052/2 I           /
000079          +                N SLIST061/2 I           /
000080          +                N SASG 064/4 I           /
000081          +                          /    61CPAL  E /
000082          +                          /    62CAP   1 /  I*
000083          +T FLT  82       T HCREW011/2 I
000084          +                T XDEST021/2*F           /
000085          +                T YDEST022/2*F 63MSK   E /
000086          +                T IDEV 031/2 I 64MASK  1 /  I*
000087          +                T ACASE032/2 I           /
000088          +                T DEP  06    F           /
000089          +                T FITON042/2 I           /
000090          +                T XI   051/2*F 65PRTM  E /
000091          +                T YI   052/2*F 66PRTSM 2 /2 F*
000092          +                T TOW  111/4 I           /
000093          +                T MFLG 112/4 I           /
000094          +                T TFLT 07    F
000095          +                T TARVL041/2 F 68PATRL E
000096          +                T RLS  08    F 69IPAT  1  4 I*
000097          +                T ROS  12    F           /
000098          +                T TOWSP112/2 F 70AVUTO 0 /  F
000099          +                               71CL    0 / *I*
000100          +                               72COSTO 0 /  I*
000101          +                               73DIST  0 /  I*
000102          +                               74EPSLN 0 /  F*
000103          +                          /    75FCQUE 0 /  I
000104          +                          /    76FCSET 0 /  I
000105          +                          /    77FEXCS 0 /  I
000106          +N ARSCH4        N LRES 0 1/2 I 78FPSET 0 /  I
000107          +                N LFLG 032/2 I 79FRQUE 0 /  I
000108          +                N LNOTE04 /  I 80FTSET 0 /  I
000109          +N ARVSN4        N CASN003 /  I 81HO    0 /  F*
000110          +                N RESN004 /  I 82IDELT 0 /  I*
000111          +N COMPL4        N NCMPL03 /  I 83IFDAY 0 /  I*
000112          +                N RCMPL04 /  I 84IMO   0 /  I
```

```
000113          +N FUEL 4        N NFUEL03 /  1 85ISA    0 /  I
000114          +                N RFUEL04 /  1 86KKA    0 /  F*
000115          +N HOME 4                  /    87LCQUE 0 /  I
000116          +                          /    88LCSET 0 /  I
000117          +N HOMEF4        N NHOM 03 /  1 89LEXCS 0 /  I
000118          +                N RHOM 04 /  1 90LPSET 0 /  I
000119          +N NUCRU4                  /    91LRQUE 0 /  I
000120          +                          /    92LTSET 0 /  I
000121          +N ONSCN4                  /    93MEANW 0 /  F
000122          +                          /    94NBRCO 0 /  I
000123          +N SNDBK4        N NSDBK03 /  1 95NBRCS 0 /  I
000124          +                N RSDBK04 /  1 96NBRFA 0 /  I
000125          +N SRISE2                  /    97NBRFB 0 /  I
000126          +                          /    98NBRFC 0 /  I
000127          +N SSET 4        N ETON 03 /  1 99NWD   0 /  I
000128          +                N SER  04 /  1100NWE   0 /  I
000129          +N STNBY4        N STAT 03 /  1101PDC1  0 /  F*
000130          +                          /   102PDC2  0 /  F*
000131          +N XSET 2                  /   103PRDN  0 /  F*
000132          +                          /   104PRUP  0 /  F*
000133          +N READY4        N NREAD031/2 1105PSHFT 0 /  I
000134          +                N RREAD032/2 1106RAP   0 /  I*
000135          +                N FREAD04    1107RISE  0 /  F*
000136          +N DELAY4                  /   108SET   0 /  F*
000137          +                          /   109SNEED 0 /  I
000138          +N CHEKN4                  /   110S2PRI 0 /  I*
000139          +                          /   111THOOK 0 /  F*
000140          +                          /   112TLAST 0 /  F
000141          +                          /   113TOSBY 0 /  I
000142          +                          /   114TOTIN 0 /  I
000143          +                          /   115TSP1  0 /  F*
000144          +                          /   116TSP2    /  F*
000145          +                          /   117TUNPR 0 /  I
000146          +                          /   118XRX   0 /  F*
000147          +                              119FLIST 0    I
000148          +                               67LLIST 0    I
000149          +                              121LIMIT 0    I
000150          +                              122KOUNT 0    I
000151          +                              123SCNVT E
000152          +                              124NSN   1 /4 I*
000153          +                              125STAPE 0    I*
000154          +                              126TPRI  0    I*
000155          +                              130TCHEK 0    F*
000156          +                              135MEAND 0    F
000157          +                              136MEANV 0    F
000158          +                              137MNTMT 0   *F
000159          +                              138GROUP E
000160          +                              139AVDRT 1  2 F
000161          +                              140CS    1  2 I
000162          +                              141FL1   1  2 I
000163          +                              142FL2   1  2 I
000164          +                              143FL3   1  2 I
000165          +                              144INTRP 1  2 I
000166          +                              145NDS   1  2 I
000167          +                              146NONPR 1  2 I
000168          +                              147NOSB  1  2 I
000169          +                              148TMTAV 1  2*F
000170          +                              149TVAVG 1  2 F
```

```
000171          +                    150JTSAVG 1  2 F
000172          +                    151USEAV  1  2 F
000173          +                    152RUTIL  0    F
000174          +                    153CUTIL  0    F
000175          +                    154C130U  0    F
000176          +                    155AIRU   0    F
000177          +                    156DSTR6  E
000178          +                    157MEEN   1  2*F
000179          +                    158CNTR   1  2 I
000180          +                    159STDEV  1  2 F
000181          +                    160CATG1  1  2 I
000182          +                    161CATG2  1  2 I
000183          +                    162CATG3  1  2 I
000184          +                    163CATG4  1  2 I
000185          +                    164CATG5  1  2 I
000186          +                    165CATG6  1  2 I
000187          +                    166CATG7  1  2 I
000188          +                    167CATG8  1  2 I
000189          +                    168CLLCT  0    F
000190          +                    169HLDY   E
000191          +                    170HOLID  1  4*I*
000192          +                    171PSTTN  1  4 I*
000193          +                    172NMBRQ  1  4 I
000194          +                    173PRTOT  0    I
000195          +                    174CFTT   1  2 F
000196          +                    175MCFTT  0    F
```

```
@ ELT EVENTS,1,710518, 49190

000001                  EVENTS
000002                     3 EXOGENOUS
000003                           OPSIM   (1)
000004                           ENDSIM  (2)
000005                           START   (3)
000006                    18 ENDOGENOUS
000007                           ARSCH
000008                           ARVSN
000009                           CHEKN
000010                           COMPL
000011                           DELAY
000012                           FUEL
000013                           HOME
000014                           HOMEF
000015                           NOTF
000016                           NOTIF
000017                           NUCRU
000018                           ONSCN
000019                           READY
000020                           SNDRK
000021                           SRISE
000022                           SSET
000023                           STNBY
000024                           XSET
000025                  END
```

```
@ ELT START,1,710604, 4904

                EXOGENOUS EVENT START
                LET TLIM = (10500.-10.*FLOAT(NRES)-19.*FLOAT(NSTA)-6.*FLOAT(NRST)
               * -6.*FLOAT(NGROUP)-5.*FLOAT(NDSTRB))/50.
                LET LIMIT = TLIM
                 LET M = 0
                 DO, FOR EACH STA I
                 DO, FOR EACH RST J
                 DO, FOR K=(1)(RFST(I,J)),
                 LET M = M + 1
                 LET STN(M) = I
                 LET TYPE(M) = J
                 LET XR(M) = XS(I)
                 LET YR(M) = YS(I)
                 LOOP
                 LOOP
                 LOOP
                 IF (M)EQ(NRES), GO TO 100
                 WRITE ON TAPE 6, M,NRES
                 FORMAT (' ERROR IN DATA  M = ',I5,' NRES = ',I5)
                 STOP
            100 LET ISA = 6-IFDAY
                 IF (ISA)LS(0), LET ISA = 6
                LET PSHFT = 1
                IF IFDAY GE 6, LET PSHFT = NWD + 1
                LET IMO = MOD(ISA+2,7)
                IF HOLID(1) EQ -1, GO TO 200
                DO, FOR EACH HLDY I
                IF (HOLID(I))EQ(0), LET PSHFT = NWD + 1
                LOOP
            200 CREATE NUCRU
                CAUSE NUCRU AT TME(PSHFT)
                LET TLAST = 0.
                 CREATE XSET
                 CAUSE XSET AT TIME + SET
                 CREATE SRISE
                 CAUSE SRISE AT TIME + RISE - XRX
                 RETURN
                 END
```

```
000001          ENDOGENOUS EVENT NUCRU                                  10000
000002        5 LET TOTME(PSHFT) = TOTME(PSHFT) + TIME-TLAST            10030
000003          DO TO 20, FOR EACH RES IRS                              10040
000004          IF EIAT(IRS) EQ 0., GO TO 10                            10050
000005          IF (TFLT(IFLT(IRS)))EQ(0.0), GO TO 10
000006          IF TFLT(IFLT(IRS)) GR TLAST, GO TO 6
000007          LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST              10070
000008          GO TO 10                                                10080
000009        6 LET TUTIL(IRS) = TUTIL(IRS) + TIME - TFLT(IFLT(IRS))    10090
000010       10 LET UTIL(IRS) = UTIL(IRS) + TUTIL(IRS)                  10100
000011          LET USHF(STN(IRS),PSHFT) = USHF(STN(IRS),PSHFT) + TUTIL(IRS)  10110
000012          LET CLLCT = CLLCT + TUTIL(IRS)
000013          LET TUTIL(IRS) = 0.                                     10120
000014       20 LOOP                                                    10130
000015           LET JJ = 0
000016           IF PSHFT EQ NWD, LET JJ = 1
000017           IF PSHFT EQ NWD+OWF, LET JJ = 2
000018           IF JJ EQ 0, GO TO 28
000019           LET MEEN(JJ) = MEEN(JJ) + CLLCT
000020           LET CNTR(JJ) = CNTR(JJ) + 1
000021           LET STDEV(JJ) = STDEV(JJ) + (CLLCT*CLLCT)
000022           LET X = (CLLCT*100.0)/FLOAT(NRES)
000023           IF X GR 0.5, GO TO 200
000024           LET CATG1(JJ) = CATG1(JJ) + 1
000025           GO TO 210
000026      200  IF X GR 1.0, GO TO 201
000027           LET CATG2(JJ) = CATG2(JJ) + 1
000028           GO TO 210
000029      201  IF X GR 2.0, GO TO 202
000030           LET CATG3(JJ) = CATG3(JJ) + 1
000031           GO TO 210
000032      202  IF X GR 3.0, GO TO 203
000033           LET CATG4(JJ) = CATG4(JJ) + 1
000034           GO TO 210
000035      203  IF X GR 4.0, GO TO 204
000036           LET CATG5(JJ) = CATG5(JJ) + 1
000037           GO TO 210
000038      204  IF X GR 5.0, GO TO 205
000039           LET CATG6(JJ) = CATG6(JJ) + 1
000040           GO TO 210
000041      205  IF X GR 10.0, GO TO 206
000042           LET CATG7(JJ) = CATG7(JJ) + 1
000043           GO TO 210
000044      206  LET CATG8(JJ) = CATG8(JJ) + 1
000045      210  LET CLLCT = 0.0
000046       28 LET I = 0
000047          LET IDAY = DPART(TIME)
000048          IF HOLID(I) EQ -1, GO TO 29
000049          DO, FOR EACH HLDY J
000050          IF IDAY EQ HOLID(J), LET I = 1
000051          LOOP
000052       29 LET IDAY = IDAY - (IDAY/7)*7
000053          IF PSHFT NE NWD, GO TO 40                               10170
000054          IF I EQ 1, GO TO 500
```

```
000055          IF IDAY NE ISA, GO TO 31                                  10180
000056      600 LET PSHFT = [illegible] + 1
000057          GO TO 70                                                  10200
000058       30 LET PSHFT = 1                                             10210
000059          GO TO 70                                                  10220
000060       40 IF PSHFT EQ NWD+NWE, GO TO 50                             10230
000061          LET PSHFT = PSHFT + 1                                     10240
000062          GO TO 70                                                  10250
000063       50 IF I EQ 1, GO TO 555
000064          IF IDAY EQ IMO, GO TO 60
000065          IF IDAY EQ ISA, GO TO 555
000066          LET ISU = 6
000067          IF IMO GR 0, LET ISU = IMO - 1
000068          IF IDAY NE ISU, GO TO 60
000069      555 LET PSHFT = NWD + 1
000070          GO TO 70                                                  10280
000071       60 LET PSHFT = 1                                             10290
000072       70 LET DD = FLOAT(DPART(TIME))                               10300
000073          CAUSE NUCRU AT DD + TME(PSHFT)                            10310
000074          LET TLAST = TIME                                          10320
000075           IF CQUE IS EMPTY, RETURN
000076           DO TO 900, FOR EACH ICNN IN CQUE
000077           IF (FLG(ICNN))EQ(2), GO TO 700
000078           IF (FLG(ICNN))EQ(3), GO TO 800
000079      600  CALL CRES(ICNN,NEED(ICNN),*KNT,*II)
000080           CALL VEC(ICNN)
000081           CALL OSET(ICNN,KNT)
000082           DO TO 650, FOR EACH IR IN RQUE
000083           IF (IB(IR))NE(0), GO TO 650
000084           IF (EIAT(IR))EQ(0.0), GO TO 610
000085           IF (HCREW(IFLT(IR)))EQ(0), GO TO 630
000086           GO TO 650
000087      610  CALL ROCA(ICNN,IR,*III)
000088           IF (III)EQ(1), GO TO 630
000089      650  LOOP
000090           GO TO 900
000091      630  LET TQUE(ICNN) = TQUE(ICNN) + (TIME-TINQ(ICNN))
000092           IF (SIGNL(ICNN))EQ(1),LET TINT(ICNN)=TINT(ICNN)+(TIME-TINQ(ICNN))
000093           REMOVE ICNN FROM CQUE
000094           CALL SERVE(ICNN,0,IR)
000095           GO TO 900
000096      700  LET CASE = CAS(ICNN)
000097           CALL CRES(CASE,NEED(ICNN),*KNT,*II)
000098           CALL VEC(CASE)
000099           CALL OSET(CASE,KNT)
000100           DO TO 750, FOR EACH IR IN RQUE
000101           IF (IB(IR))NE(0), GO TO 750
000102           IF (EIAT(IR))EQ(0.0), GO TO 710
000103           IF (HCREW(IFLT(IR)))EQ(0), GO TO 730
000104           GO TO 750
000105      710  CALL ROCA(CASE,IR,*III)
000106           IF (III)EQ(1), GO TO 730
000107      750  LOOP
000108           GO TO 900
000109      730  CALL SVQUE(ICNN,IR)
000110           GO TO 900
000111      800  LET CASE = FSAC(ICNN)
000112           CALL SASS(CASE,SM(ICNN),ICNN,*IOK)
```

```
000113          GO TO 850, FOR EACH IR IN RQUE
000114          IF (IR(IR))EQ(0), GO TO 850
000115          IF (EIAT(IR))EQ(0.0), GO TO 81
000116          IF (BCREM(IR,LT(IR)))EQ( ), GO TO 830
000117          GO TO 850
000118     810  CALL ROCA(CASE,IR,*IJK)
000119          IF (IJK)EQ(1), GO TO 830
000120          GO TO 850
000121     830  IF (VOP(TYPE(IR)))EQ(0), GO TO 870
000122          LET X = AINT(TIME) + RISE
000123          IF (TIME)LE(X), GO TO 870
000124          LET X = AINT(TIME) + SET
000125          IF (TIME)GR(X), GO TO 870
000126          LET DVEC(IR) = TVEC(IR)
000127          IF EIAT(IR) EQ 0.0, LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000128          IF (DVEC(IR))LS(X-TIME), GO TO 870
000129          LET X = AINT(TIME) + RISE + 1.0
000130          IF (TIME+DVEC(IR))GE(X), GO TO 870
000131          IF (EIAT(IR))EQ(0.0), LET BCREM(STN(IR))=BCREM(STN(IR))-1
000132     850  LOOP
000133          GO TO 900
000134     870  LET I = SIGNL(ICNN)
000135          LET SIGNL(ICNN) = 2
000136          LET JIND = 2
000137          LET KIND = 2
000138          DO TO 872, FOR EACH J IN SRHS(CASE)
000139          IF SIGNL(J) EQ 0, LET JIND = 1
000140          IF SIGNL(J) EQ 1, LET KIND = 1
000141     872  LOOP
000142          IF (I)EQ(1), GO TO 878
000143          IF (JIND)EQ(1), GO TO 877
000144          IF (KIND)EQ(1), GO TO 877
000145          LET TQUE(CASE) = TQUE(CASE)+(TIME-TINQ(CASE))
000146          GO TO 877
000147     878  IF (KIND)EQ(1), GO TO 877
000148          IF (JIND)EQ(2), GO TO 866
000149          LET TINT(CASE) = TINT(CASE)+(TIME-STINQ(CASE))
000150          GO TO 877
000151     866  LET TQUE(CASE) = TQUE(CASE) + (TIME-TINQ(CASE))
000152          LET TINT(CASE) = TINT(CASE) + (TIME-STINQ(CASE))
000153     877  LET RSRC(ICNN) = IR
000154          LET SASG(ICNN) = 1
000155          REMOVE ICNN FROM CQUE
000156          CALL SSS(ICNN,IR,3)
000157     900  LOOP
000158          RETURN
000159          END
```

```
000001                EXECUTE SEQ FOR OPSI
000002          C
000003          C     OPSIM CREATES AN ENTITY CASE, READS IN CASE ATTRIBUTES FROM THE
000004          C     DEMAND TAPE, CHECKS FOR IMPOSSIBLE COMBINATIONS OF CASE ATTRIBUTES,
000005          C     AND DETERMINES WHICH SUBROUTINE TO CALL.
000006          C
000007                SAVE EVENT CARD
000008                LET KOUNT = KOUNT + 1
000009                LET DUM = 9999.0
000010                CREATE CASE
000011                LET NBRCS = NBRCS + 1
000012                READ IDLOC(CASE),STATN(CASE),NOCAS(CASE),FOR(CASE),AIR(CASE),
000013               * WIND(CASE),SWELL(CASE),VIS(CASE),CTYPE(CASE),L(CASE),FPRI(CASE),
000014                FORMAT (I3,11I5)
000015                READ NN(CASE),MMM(CASE),IDUM,S1S(CASE),S2S(CASE),JDUM,LDUM,IX,
000016               * IY,BOX(CASE),VALUE(CASE)
000017                FORMAT (10I5,I10)
000018                IF PRTOT EQ 0, GO TO 4
000019                WRITE ON TAPE 6, CASE,TIME
000020                FORMAT (' CASE CREATED IS ',I6,' AT TIME ',M3.2.2)
000021                WRITE ON TAPE 6, IDLOC(CASE),STATN(CASE),NOCAS(CASE),AIR(CASE),
000022               * WIND(CASE),SWELL(CASE),VIS(CASE),FPRI(CASE),N,NN(CASE),
000023               * MMM(CASE),S1S(CASE),S2S(CASE),JDUM,LDUM,IX,IY
000024                FORMAT (I3,I3,I5,4I4,6I3,4I5)
000025              4 IF MMM(CASE) GR 2, LET MMM(CASE) = 2
000026                LET OPFAC(CASE) = STATN(CASE)
000027                LET XCX(CASE) = FLOAT(IX)
000028                LET YCY(CASE) = FLOAT(IY)
000029                LET XC(CASE) = XCX(CASE)
000030                LET YC(CASE) = YCY(CASE)
000031                IF (STATN(CASE))EQ(0), GO TO 3
000032                LET STATN(CASE) = NSN(STATN(CASE))
000033                IF (STATN(CASE))EQ(0), GO TO 3
000034                IF (XS(STATN(CASE)))NE(0.0), GO TO 10
000035                IF (YS(STATN(CASE)))NE(0.0), GO TO 10
000036              3 DO TO 5, FOR EACH STA I, WITH (PSTTN(I))EQ(1)
000037                IF (XS(I))NE(0.0), GO TO 1
000038                IF (YS(I))NE(0.0), GO TO 1
000039                GO TO 5
000040              1 LET X = XS(I) - XC(CASE)
000041                LET Y = YS(I) - YC(CASE)
000042                LET D = SQRT(X*X+Y*Y)
000043                IF (D)GE(DUM), GO TO 5
000044                LET DUM = D
000045                LET STATN(CASE) = I
000046              5 LOOP
000047                GO TO 20
000048             10 IF (PSTTN(STATN(CASE)))EQ(0), GO TO 3
000049                LET X = XS(STATN(CASE))-XC(CASE)
000050                LET Y = YS(STATN(CASE))-YC(CASE)
000051                LET DUM = SQRT(X*X+Y*Y)
000052                LET JJ = STATN(CASE)
000053                IF (ADJS(STATN(CASE),1))EQ(0), GO TO 20
000054                DO TO 15, FOR I = (1)(NADJS(STATN(CASE)))
```

```
000055          IF (XS(ADJS(STATN(CASE),I)))NE(0.0), GO TO 2
000056          IF (YS(ADJS(STATN(CASE),I)))NE(0.0), GO TO 2
000057          GO TO 15
000058        2 IF (PSTTN(ADJS(STATN(CASE),I)))EQ(0), GO TO 15
000059          LET X = XS(ADJS(STATN(CASE),I)) - XC(CASE)
000060          LET Y = YS(ADJS(STATN(CASE),I)) - YC(CASE)
000061          LET D = SQRT(X*X+Y*Y)
000062          IF (D)GE(DUM), GO TO 15
000063          LET DUM = D
000064          LET JJ = ADJS(STATN(CASE),I)
000065       15 LOOP
000066          LET STATN(CASE) = JJ
000067       20 LET GAMMA(CASE) = FLOAT(IDUM)/100.0
000068          LET TSM(CASE) = FLOAT(JDUM)
000069          LET OFSHR(CASE) = FLOAT(LDUM)/10.0
000070     C
000071     C    INITIALIZE CASE ATTRIBUTES
000072     C
000073          LET COSTC(CASE) = 0.0
000074          LET FLG(CASE) = 0
000075          IF (N)EQ(1), LET FLG(CASE) = 1
000076          LET NEED(CASE) = 0
000077          LET ITOL(CASE) = 2
000078          LET IWAIT(CASE) = 0
000079          LET LOC(CASE) = 0
000080          LET NOINT(CASE) = 0
000081          LET NQUE(CASE) = 0
000082          LET CNRES(CASE) = 0
000083          LET OCCUR(CASE) = TIME
000084          LET PRI(CASE) = FPRI(CASE)
000085          LET REA(CASE) = 2
000086          LET RESA(CASE) = 0
000087          LET SYTAG(CASE) = 0
000088          LET TINT(CASE) = 0.0
000089          LET TQUE(CASE) = 0.0
000090          LET TQUE1(CASE) = 0.0
000091          LET TSVC(CASE) = 0.0
000092        LET IS2(CASE) = 0
000093          IF (N)LE(0), GO TO 500
000094          IF (N)GR(1), GO TO 300
000095          IF (MMM(CASE))GR(1), GO TO 300
000096     C
000097     C    READ IN NEED OF SINGLE RESOURCE CASE.
000098     C
000099          READ NEED(CASE),OST(CASE)
000100          FORMAT (I2,D1.4)
000101          IF PRTOT EQ 0, GO TO 500
000102          WRITE ON TAPE 6, NEED(CASE),OST(CASE),OST(CASE)
000103          FORMAT (' NEED ',I2,' OST ',D1.4,' OR ',M3.2.2)
000104          GO TO 500
000105     C
000106     C    CREATE NOTIFS FOR MULTI-RESOURCE NEEDS; READ IN NEED(I), OST(I),
000107     C    DELTA(I) FOR EACH NOTIF.
000108     C
000109      300 IF (NNN(CASE))EQ(0), GO TO 400
000110          LET INI = N
000111          IF (MMM(CASE))GR(0), LET INI = N - 1
000112          DO, FOR I = (1)(INI)
```

```
000113                CREATE NOTIF
000114                STORE CASE IN CASE(NTI+1)
000115                LET FLG(NOTIF) = 2
000116                LET PRI(NOTIF) = PRI(CASE)
000117                LET RESA(NOTIF) = 0
000118                LET KRES(NOTIF) = 0
000119                LET NUMBR(NOTIF) = 0
000120               LET SIGNL(NOTIF) = 2
000121                FILE NOTIF IN NSET(CASE)
000122                LOOP
000123                READ NEED(I),OST(I),DELTA(I), FOR EACH I IN NSET(CASE)
000124                FORMAT 6(I2,D1.4,D1.2)
000125                IF PRTOT EQ 0, GO TO 777
000126                DO, FOR EACH I IN NSET(CASE)
000127                WRITE ON TAPE 6, NEED(I),OST(I),DELTA(I)
000128                FORMAT (' NEED ',I2,' OST ',M3.2.2,' DELTA ',D1.2)
000129                LOOP
000130           777  IF (MMM(CASE))LE(0), GO TO 500
000131           400  READ ITOW(CASE)
000132                FORMAT (I2)
000133                IF PRTOT EQ 0, GO TO 888
000134                WRITE ON TAPE 6, ITOW(CASE)
000135                FORMAT (' ITOW ',I2)
000136          C
000137          C     CHECK FOR IMPOSSIBLE COMBINATIONS OF CASE ATTRIBUTES.
000138          C
000139           888  IF ITOW(CASE) NE 17, GO TO 500
000140                IF NNN(CASE) EQ 0, GO TO 500
000141           200  IF NSET(CASE) IS EMPTY, GO TO 210
000142                REMOVE FIRST NOTIF FROM NSET(CASE)
000143                DESTROY NOTIF
000144                GO TO 200
000145           210  LET ITOL(CASE) = 6
000146                FILE CASE IN EXCS
000147                RETURN
000148           500 IF KOUNT LS LIMIT, GO TO 501
000149               IF LIMIT NE 0, WRITE ON TAPE 6, TIME,LIMIT
000150               FORMAT (' W A R N I N G      AT TIME',M4.2.2,' THERE ARE AT LEAST,
000151              *,I5,' CASES IN THE SYSTEM.'/' NO MORE CASES WILL OCCUR. THOSE ALRE
000152              *ADY IN THE SYSTEM WILL RUN TO COMPLETION.')
000153               LET LIMIT = 0
000154           502 IF NSET(CASE) IS EMPTY, GO TO 503
000155               REMOVE FIRST NOTIF FROM NSET(CASE)
000156               DESTROY NOTIF
000157               GO TO 502
000158           503 DESTROY CASE
000159               RETURN
000160           501 LET I = NNN(CASE)+MMM(CASE)+S1S(CASE)+S2S(CASE)
000161                IF (I)NE(0), GO TO 30
000162                GO TO 700
000163            30  IF (S1S(CASE))EQ(0), GO TO 31
000164                IF (S2S(CASE))EQ(0), GO TO 31
000165                GO TO 700
000166            31  LET I = NNN(CASE)+MMM(CASE)+S1S(CASE)
000167                IF (I)NE(0), GO TO 600
000168           700  LET ITOL(CASE) = 6
000169                FILE CASE IN EXCS
000170                IF (N)LE(I), GO TO 701
```

```
000171          702  IF MSET(CASE) IS EMPTY, GO TO 701
000172               REMOVE FIRST NOTIF FROM SET(CASE)
000173               DESTROY NOTIF
000174               GO TO 702
000175          701  RETURN
000176        C
000177        C      EXAMINE INPUT PARAMETERS TO DETERMINE WHICH SUBROUTINE TO CALL
000178        C
000179          600  IF (SIS(CASE))EQ(0), GO TO 520
000180               CALL SRCH(CASE)
000181               RETURN
000182          520  LET I = NNN(CASE) + AMI(CASE)
000183               IF (I)EQ(1), GO TO 530
000184               CALL MRAS(CASE)
000185               RETURN
000186          530  CALL SRAS(CASE)
000187               RETURN
000188               END
```

```
      SUBROUTINE GRAS(CASE)
C
C     SUBROUTINE GRAS SERVES AS A DRIVER FOR SINGLE RESOURCE CASES.
C
      CALL CRES(CASE,NEED(CASE),*KNT,*II)
      IF II NE 0, GO TO 20
      LET ITOL(CASE) = 3
      LET FAIL1(STATN(CASE)) = FAIL1(STATN(CASE)) + 1
      FILE CASE IN EXCS
      RETURN
   20 IF KNT NE 0, GO TO 30
      LET ITOL(CASE) = 4
      LET FAIL2(STATN(CASE)) = FAIL2(STATN(CASE)) + 1
      FILE CASE IN EXCS
      RETURN
   30 CALL VEC(CASE)
      CALL OSET(CASE,KNT)
      CALL RESAP(CASE,RAP,*IRES)
      IF (IRES)EQ(0), GO TO 100
      CALL SERVE(CASE,0,IRES)
   50 RETURN
  100 IF (NEED(CASE))EQ(17), GO TO 200
      LET TINQ(CASE) = TIME
      IF (REA(CASE))EQ(2), LET REA(CASE) = 1
      LET NQUE(CASE) = NQUE(CASE) + 1
      LET SIGNL(CASE) = 0
      FILE CASE IN CQUE
      RETURN
  200 LET ITOL(CASE) = 5
      FILE CASE IN EXCS
      RETURN
      END
```

```
000001                SUBROUTINE CRES(CASE,I ,KO T,II)
000002          C
000003          C     SUBROUTINE CRES DETERMINES CAPABLE RESOURCE TYPES AND DETERMI S
000004          C     ALL STATIONS FROM WHICH RESOURCES CAN BE OBTAINED.  IT THEN
000005          C     PLACES ALL CAPABLE RESOURCES MEETING THE TYPE AND STATION
000006          C     REQUIREMENTS INTO A FIFO SET--RQUE.
000007          C
000008                DIMENSION IRAY1(20)
000009                DIMENSION IRAY2(30)
000010                LET J = 0
000011                LET KONTR = 0
000012                IF RQUE IS EMPTY, GO TO 5
000013             10 REMOVE FIRST IRS FROM RQUE
000014                IF RQUE IS NOT EMPTY, GO TO 10
000015              5 IF (SMELL(CASE))GE(5), GO TO 20
000016                LET IRAY1(2)=28
000017                GO TO 30
000018             20 IF (SMELL(CASE))GE(10), GO TO 21
000019                LET IRAY1(2)=29
000020                GO TO 30
000021             21 IF (SMELL(CASE))GE(20), GO TO 22
000022                LET IRAY1(2)=30
000023                GO TO 30
000024             22 LET IRAY1(2)=31
000025             30 IF (WIND(CASE))LS(60), GO TO 31
000026                LET IRAY1(3)=33
000027                GO TO 32
000028             31 LET IRAY1(3)=32
000029             32 IF (VIS(CASE))GR(0), GO TO 40
000030                LET IRAY1(4)=34
000031                GO TO 60
000032             40 LET IRAY1(4)=35
000033             60 IF (AIR(CASE))LS(20), GO TO 61
000034                LET IRAY1(6)=37
000035                GO TO 62
000036             61 LET IRAY1(6)=36
000037             62 IF (OFSHR(CASE))GE(.5), GO TO 70
000038                LET IRAY1(5)=38
000039                GO TO 80
000040             70 IF (OFSHR(CASE))GE(5.0), GO TO 71
000041                LET IRAY1(5)=39
000042                GO TO 80
000043             71 IF (OFSHR(CASE))GE(10.0), GO TO 72
000044                LET IRAY1(5)=40
000045                GO TO 80
000046             72 IF (OFSHR(CASE))GE(20.0), GO TO 73
000047                LET IRAY1(5)=41
000048                GO TO 80
000049             73 IF (OFSHR(CASE))GE(50.0), GO TO 74
000050                LET IRAY1(5)=42
000051                GO TO 80
000052             74 LET IRAY1(5)=43
000053             80 IF (IND)EQ(15), GO TO 81
000054                IF (IND)E.(16), GO TO 9
```

```
000055          LET IRAY1(1)=I
000056          GO TO 100
000057       81 IF (L(CASE))GE(30), GO TO 82
000058          LET IRAY1(1)=19
000059          GO TO 100
000060       82 IF (L(CASE))GE(45), GO TO 83
000061          LET IRAY1(1)=20
000062          GO TO 100
000063       83 IF (L(CASE))GE(100), GO TO 84
000064          LET IRAY1(1)=21
000065          GO TO 100
000066       84 IF (L(CASE))GE(200), GO TO 85
000067          LET IRAY1(1)=22
000068          GO TO 100
000069       85 LET IRAY1(1)=23
000070          GO TO 100
000071       90 IF (POB(CASE))GE(5), GO TO 91
000072          LET IRAY1(1)=16
000073          GO TO 100
000074       91 IF (POB(CASE))GE(10), GO TO 92
000075          LET IRAY1(1)=24
000076          GO TO 100
000077       92 IF (POB(CASE))GE(18), GO TO 93
000078          LET IRAY1(1)=25
000079          GO TO 100
000080       93 IF (POB(CASE))GE(25), GO TO 94
000081          LET IRAY1(1)=26
000082          GO TO 100
000083       94 LET IRAY1(1)=27
000084      100 LET IAND = CAP(IRAY1(1))
000085          DO TO 101, FOR I=(2)(6)
000086          STORE AND(CAP(IRAY1(I)),IAND) IN IAND
000087      101 LOOP
000088          IF (IAND)GR(0), GO TO 102
000089          LET KONTR = 0
000090          LET II = 0
000091          RETURN
000092      102 LET K=0
000093          DO TO 103, FOR I=(1)(NRST)
000094          STORE AND(IAND,MASK(I)) IN ITEMP
000095          IF (ITEMP)LE(0), GO TO 103
000096          LET K=K+1
000097          LET IRAY1(K)=I
000098      103 LOOP
000099          LET II=1
000100          IF (SIS(CASE))GR(0), GO TO 210
000101          IF (MMM(CASE)+NNN(CASE))GR(0), GO TO 210
000102          GO TO (210,210,210,201,201,210), RAP
000103      201 LET IRAY2(1) = STATN(CASE)
000104          LET J=J+1
000105          DO TO 202, FOR I=(1)(NACS(STATN(CASE)))
000106          IF (ACS(STATN(CASE),I))EQ(0), GO TO 204
000107          LET J=J+1
000108          LET IRAY2(J) = ACS(STATN(CASE),I)
000109      202 LOOP
000110      204 DO TO 203, FOR I=(1)(NCUT(STATN(CASE)))
000111          IF (CUT(STATN(CASE),I))EQ(0), GO TO 205
000112          LET J=J+1
```

```
000113                LET IRAY2(J) = CUT(STAT (CASE),I)
000114          203   LOOP
000115          205   GO TO 250
000116          210   LET IRAY2(1) = STATN(CASE)
000117                LET J=J+1
000118                DO TO 211, FOR I=(1)(NADJS(STATN(CASE)))
000119                IF (ADJS(STATN(CASE),I))EQ(0), GO TO 219
000120                LET J=J+1
000121                LET IRAY2(J) = ADJS(STATN(CASE),I)
000122          211   LOOP
000123          219   DO TO 212, FOR I=(1)(NACS(STATN(CASE)))
000124                IF (ACS(STATN(CASE),I))EQ(0), GO TO 206
000125                LET J=J+1
000126                LET IRAY2(J) = ACS(STATN(CASE),I)
000127          212   LOOP
000128          206   DO TO 213, FOR I=(1)(NCUT(STATN(CASE)))
000129                IF (CUT(STATN(CASE),I))EQ(0), GO TO 207
000130                LET J=J+1
000131                LET IRAY2(J) = CUT(STATN(CASE),I)
000132          213   LOOP
000133          207   DO TO 214, FOR I=(1)(NADJS(STATN(CASE)))
000134                IF (ADJS(STATN(CASE),I))EQ(0), GO TO 250
000135                DO TO 215, FOR LL = (1)(NACS(ADJS(STATN(CASE),I)))
000136                IF (ACS(ADJS(STATN(CASE),I),LL))EQ(0), GO TO 208
000137                LET NO = ACS(ADJS(STATN(CASE),I),LL)
000138                LET ITAG=0
000139                DO TO 216, FOR M=(1)(J)
000140                IF (NO)NE(IRAY2(M)), GO TO 216
000141                LET ITAG=1
000142          216   LOOP
000143                IF (ITAG)EQ(1), GO TO 215
000144                LET J=J+1
000145                LET IRAY2(J)=NO
000146          215   LOOP
000147          208   DO TO 217, FOR LL=(1)(NCUT(ADJS(STATN(CASE),I)))
000148                IF (CUT(ADJS(STATN(CASE),I),LL))EQ(0), GO TO 214
000149                LET NO = CUT(ADJS(STATN(CASE),I),LL)
000150                LET ITAG=0
000151                DO TO 218, FOR M=(1)(J)
000152                IF (NO)NE(IRAY2(M)), GO TO 218
000153                LET ITAG=1
000154          218   LOOP
000155                IF (ITAG)EQ(1), GO TO 217
000156                LET J=J+1
000157                LET IRAY2(J)=NO
000158          217   LOOP
000159          214   LOOP
000160          250   DO TO 260, FOR EACH RES I
000161                DO TO 270, FOR LL=(1)(J,
000162                IF (STN(I))EQ(IRAY2(LL)), GO TO 280
000163          270   LOOP
000164                GO TO 260
000165          280   DO TO 290, FOR LL=(1)(K)
000166                IF (TYPE(I))EQ(IRAY1(LL)), GO TO 291
000167          290   LOOP
000168                GO TO 260
000169          291   FILE I IN RCDE
000170                LET KCUTR=KCNTR+1
```

```
000171          260  LOOP
000172               SET
000173               END
```

@ ELT VEC,1,71C5,P, 69131

```
                SUBROUTINE  EC(CASE)
        C
        C       SUBROUTINE VEC CALCULATES TIME-TO-VECTOR FOR EACH RES IN RQUE.
        C
     10 DO TO 100, FOR EACH IRS OF RQUE
        IF (EIAT(IRS))EQ(0.0), GO TO 15
        IF (DEP(IFLT(IRS)))NE(0.0), GO TO 20
     15 LET X = XR(IRS)-XC(CASE)
        LET Y = YR(IRS)-YC(CASE)
        LET D = SQRT(X*X+Y*Y)
        IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 30
        LET TVEC(IRS) = D/SOA1(TYPE(IRS))
        GO TO 100
     30 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
        GO TO 100
     20 LET X = XR(IRS)-XDEST(IFLT(IRS))
        LET Y = YR(IRS)-YDEST(IFLT(IRS))
        LET D = SQRT(X*X+Y*Y)
        IF (IR(IRS))GE(1), GO TO 60
        IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 61
        LET C = (TIME-DEP(IFLT(IRS)))*SOA1(TYPE(IRS))
        GO TO 70
     61 LET C = (TIME-DEP(IFLT(IRS)))*SOA2(TYPE(IRS))
        GO TO 70
     60 IF (TOW(IFLT(IRS)))EQ(0), GO TO 40
        LET C = TIME - DEP(IFLT(IRS))
        IF (C)LE(0.0), GO TO 65
        LET C = (TIME-DEP(IFLT(IRS)))*TOWSP(IFLT(IRS))
        GO TO 70
     65 LET C = 0.0
        GO TO 70
     40 IF (SWELL(ACASE(IFLT(IRS))))GR(SLIM(TYPE(IRS))), GO TO 50
        LET C = (TIME-DEP(IFLT(IRS)))*SOA1(TYPE(IRS))
        GO TO 70
     50 LET C = (TIME-DEP(IFLT(IRS)))*SOA2(TYPE(IRS))
     70 IF (C)GE(D), GO TO 80
        LET XI(IFLT(IRS)) = XR(IRS)-((C*X)/D)
        LET YI(IFLT(IRS)) = YR(IRS)-((C*Y)/D)
        GO TO 90
     80 LET XI(IFLT(IRS)) = XDEST(IFLT(IRS))
        LET YI(IFLT(IRS)) = YDEST(IFLT(IRS))
     90 LET X = XI(IFLT(IRS))-XC(CASE)
        LET Y = YI(IFLT(IRS))-YC(CASE)
        LET D = SQRT(X*X+Y*Y)
        IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 95
        LET TVEC(IRS) = D/SOA1(TYPE(IRS))
        GO TO 100
     95 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
    100 REPEAT 10
        RETURN
        END
```

```
@ ELT OSET,1,710G1R, 69131

000001                SUBROUTINE OSET(CASE,KONTR)
000002          C
000003          C     OSET RANKS THE CAPABLE RESOURCES,WITH TVEC(RES) LESS THAN OR
000004          C     EQUAL TO TOL(I),IN INCREASING COST AND RANKS THE REMAINING
000005          C     RESOURCES IN INCREASING TVEC(RES). THE TAG SETS ARE THEN COMBINE-
000006          C     INTO A SINGLE LIST.
000007          C
000008                DO TO 100, FOR I=(1)(KONTR)
000009                REMOVE FIRST IRS FROM RQUE
000010                LET DVEC(IRS) = TVEC(IRS)
000011                IF (EIAT(IRS))EQ(0.0), LET DVEC(IRS)=TVEC(IRS)+DLAY(TYPE(IRS))
000012                IF (DVEC(IRS))LE(TOL(PRI(CASE))), GO TO 20
000013                FILE IRS IN TSET
000014                GO TO 100
000015             20 IF (COSTD)EQ(0), GO TO 30
000016                LET COST(IRS)=COSTD(TYPE(IRS))*TVEC(IRS)
000017                FILE IRS IN CSET
000018                GO TO 100
000019             30 LET COST(IRS) = RCOST(TYPE(IRS))
000020                FILE IRS IN CSET
000021            100 LOOP
000022             35 IF CSET IS EMPTY, GO TO 36
000023                REMOVE FIRST IRS FROM CSET
000024                FILE IRS IN RQUE
000025                GO TO 35
000026             36 IF TSET IS EMPTY, GO TO 37
000027                REMOVE FIRST IRS FROM TSET
000028                FILE IRS IN RQUE
000029                GO TO 36
000030             37 DO TO 40, FOR EACH III IN RQUE
000031                LET COST(III) = COSTD(TYPE(III))*TVEC(III)
000032             40 LOOP
000033                RETURN
000034                END
```

```
> ELT RESAP,1,71-519, A370

      SUBROUTINE  RESAP(CASE,IRAP,JRS)
C
C     SUBROUTINE RESAP SETS UP THE PROPER SEQUENCE OF SEARCHING FOR
C     IDLE RESOURCES AND RESOURCES THAT CAN BE INTERRUPTED AT THE
C     PRIMARY AND ADJACENT STATIONS. THIS PROPER SEQUENCE IS DETERMINE
C     BY THE USER SELECTED VALUE OF RAP(RESOURCE ASSIGNMENT POLICY).
C
      GO TO (100,100,200,200,200,100), IRAP
C
C     SEARCH FOR AN IDLE RESOURCE AT THE PRIMARY STATION FOR RAP=1,2,6
C
  100 DO TO 110, FOR EACH IRS IN RQUE
      IF (IB(IRS))NE(0), GO TO 110
      IF (STN(IRS))EQ(STATN(CASE)), GO TO 150
      DO TO 121, FOR I=(1)(NACS(STATN(CASE)))
      IF (STN(IRS))EQ(ACS(STATN(CASE),I)),  GO TO 150
  121 LOOP
      DO TO 123, FOR I=(1)(NCUT(STATN(CASE)))
      IF (STN(IRS))EQ(CUT(STATN(CASE),I)), GO TO 150
  123 LOOP
      GO TO 110
  150 IF (ETAT(IRS))EQ(0.0), GO TO 152
      IF (HCREW(IFLT(IRS)))GE(1), GO TO 110
      LET JRS = IRS
  130 RETURN
  152 CALL ROCA(CASE,IRS,*IVAL)
      IF (IVAL)EQ(2), GO TO 110
      LET JRS = IRS
      RETURN
  110 LOOP
      GO TO (300,400,900,900,900,300), IRAP
C
C     SEARCH FOR AN IDLE RESOURCE IN RQUE FOR RAP = 3,4,5
C
  200 DO TO 210, FOR EACH IRS OF RQUE
      IF (IB(IRS))NE(0), GO TO 210
      IF (ETAT(IRS))EQ(0.0), GO TO 202
      IF (HCREW(IFLT(IRS)))GE(1), GO TO 210
      LET JRS = IRS
  203 RETURN
  202 CALL ROCA(CASE,IRS,*IVAL)
      IF (IVAL)EQ(2), GO TO 210
      LET JRS = IRS
      RETURN
  210 LOOP
      GO TO (900,900,700,700,800,900), IRAP
C
C     SEARCH FOR AN IDLE RESOURCE AT AN ADJACENT STATION FOR RAP=1,2,6
C
  300 IF (ADJS(STATN(CASE),1))EQ(0), GO TO 320
      DO TO 310, FOR EACH IRS OF RQUE
      IF (IB(IRS))NE(0), GO TO 310
      DO TO 311, FOR I=(1)(NADJS(STATN(CASE)))
      IF (STN(IRS))EQ(ADJS(STATN(CASE),I)), GO TO 353
```

```
000055                GO TO 312, FOR K=(1)(NACS(ADJS(STATN(CASE),I)))
000056                IF (STN(IRS))EQ(ACS(ADJS(STATN(CASE),I),K)), GO TO 311
000057           312  LOOP
000058                GO TO 313, FOR K=(1)(NCUT(ADJS(STATN(CASE),I)))
000059                IF (STN(IRS))EQ(CUT(ADJS(STATN(CASE),I),K)), GO TO 311
000060           313  LOOP
000061           311  LOOP
000062                GO TO 310
000063           350  IF (ELAT(IRS))EQ(0.0), GO TO 352
000064                IF (HCREA(IFLT(IRS)))GE(1), GO TO 310
000065                LET JRS = IRS
000066           353  RETURN
000067           352  CALL ROCA(CASE,IRS,*IVAL)
000068                IF (IVAL)EQ(2), GO TO 310
000069                LET JRS = IRS
000070                RETURN
000071           310  LOOP
000072           320  GO TO (400,500,900,900,900,800), IRAP
000073      C
000074      C         SEARCH FOR A RESOURCE THAT CAN BE INTERRUPTED AT THE PRIMARY
000075      C         STATION FOR RAP=1,2
000076      C
000077           400  LET IDUM = 100
000078                DO TO 410, FOR EACH IRS OF RQUE
000079                IF (IB(IRS))EQ(0), GO TO 410
000080                IF (STN(IRS))EQ(STATN(CASE)), GO TO 450
000081                DO TO 421, FOR I=(1)(NACS(STATN(CASE)))
000082                IF (STN(IRS))EQ(ACS(STATN(CASE),I)), GO TO 450
000083           421  LOOP
000084                DO TO 423, FOR I=(1)(NCUT(STATN(CASE)))
000085                IF (STN(IRS))EQ(CUT(STATN(CASE),I)), GO TO 450
000086           423  LOOP
000087                GO TO 410
000088           450  IF (HCREA(IFLT(IRS)))GE(0), GO TO 410
000089                IF (PRIOR(IRS))GE(PRI(CASE)),GO TO 410
000090                IF (PRIOR(IRS))GE(IDUM), GO TO 410
000091                LET IDUM = PRIOR(IRS)
000092                LET JRS = IRS
000093           410  LOOP
000094                IF (IDUM)EQ(100), GO TO 460
000095           488  LET X = TVEC(JRS)
000096                LET Y = COST(JRS)
000097                LET Z = DVEC(JRS)
000098                CALL SAD(JRS)
000099                LET TVEC(JRS) = X
000100                LET COST(JRS) = Y
000101                LET DVEC(JRS) = Z
000102                RETURN
000103           460  GO TO (500,300,900,900,900,900), IRAP
000104      C
000105      C         SEARCH FOR A RESOURCE THAT CAN BE INTERRUPTED AT THE ADJACENT
000106      C         STATIONS FOR RAP=1,2
000107      C
000108           500  IF (ADJS(STATN(CASE),1))EQ(0), GO TO 560
000109                LET IDUM = 100
000110                DO TO 510, FOR EACH IRS IN RQUE
000111                IF (IB(IRS))EQ(0), GO TO 510
000112                IF (HCREA(IFLT(IRS)))GE(0), GO TO 510
```

```
000113                DO TO 511, FOR I=(1)(NADJS(STATN(CASE)))
000114                IF (STN(IRS))EQ(ADJS(STATN(CASE),I)), GO TO 550
000115                DO TO 512, FOR K=(1)(NACS(ADJS(STATN(CASE),I)))
000116                IF (STN(IRS))EQ(ACS(ADJS(STATN(CASE),I),K)), GO TO 550
000117           512  LOOP
000118                DO TO 513, FOR K=(1)(NCUT(ADJS(STATN(CASE),I)))
000119                IF (STN(IRS))EQ(CUT(ADJS(STATN(CASE),I),K)), GO TO 550
000120           513  LOOP
000121           511  LOOP
000122                GO TO 510
000123           550  IF (PRIOR(IRS))GE(PRI(CASE)), GO TO 510
000124                IF (PRIOR(IRS))GE(IDUM), GO TO 510
000125                LET IDUM = PRIOR(IRS)
000126                LET JRS = IRS
000127           510  LOOP
000128                IF (IDUM)NE(100), GO TO 488
000129           560  GO TO (800,800,900,900,900,900), IRAP
000130      C
000131      C         SEARCH FOR A RESOURCE THAT CAN BE INTERRUPTED IN RQUE FOR RAP=3,4
000132      C
000133           700  LET IDUM = 100
000134                DO TO 710, FOR EACH IRS ON RQUE
000135                IF (IB(IRS))EQ(0), GO TO 710
000136                IF (HCREW(IFLT(IRS)))GR(0), GO TO 710
000137                IF (PRIOR(IRS))GE(PRI(CASE)), GO TO 710
000138                IF (PRIOR(IRS))GE(IDUM), GO TO 710
000139                LET IDUM = PRIOR(IRS)
000140                LET JRS = IRS
000141           710  LOOP
000142                IF (IDUM)NE(100), GO TO 488
000143                GO TO (900,900,800,800,900,900), IRAP
000144      C
000145      C         RETURN WITH RES=0 SINCE NO RESOURCE CAN BE ASSIGNED FOR
000146      C         RAP=1,2,3,4,5,6
000147      C
000148           800  LET JRS = 0
000149                RETURN
000150           900  WRITE ON TAPE 6, NOCAS(CASE)
000151                FORMAT (' SOMETHING JOLLY WELL WENT HAYWIRE WHEN CASE ',I8,
000152              1 ' ENTERED THE SYSTEM.'/)
000153                RETURN
000154                END
```

```
@ ELT ROCA,1,710519, 63730

      SUBROUTINE ROCA(CASE,IRS,II)
C
C     SUBROUTINE ROCA DETERMINES THE OPERATIONAL STATUS AND THE
C     AVAILABILITY OF A CREW FOR A RESOURCE.
C
      IF (RANDM)LE(MR(TYPE(IRS))), GO TO 10
      LET II = 2
      RETURN
   10 LET K = SHIFT(STN(IRS),PSHFT)-BCREW(STN(IRS))
      IF (CL)GE(0), GO TO 20
      IF (K)GR(0), GO TO 15
      LET II = 2
      RETURN
   20 IF (K)LE(CL), GO TO 30
   15 LET II = 1
      LET BCREW(STN(IRS))=BCREW(STN(IRS))+1
      RETURN
   30 IF (K)LS(CL), GO TO 50
      IF (CL)EQ(0), GO TO 50
      LET II = 1
      LET BCREW(STN(IRS))=BCREW(STN(IRS))+1
      GO TO 40
   50 LET II = 2
   40 IF (SYTAG(CASE))EQ(1), GO TO 60
      LET SYTAG(CASE) = 1
      CREATE STNBY
      LET NSTBY(STN(IRS)) = NSTBY(STN(IRS)) + 1
      LET TOSBY = TOSBY + 1
      STORE STN(IRS) IN STAT(STNBY)
      CAUSE STNBY AT TIME + (2.0/24.0)
   60 RETURN
      END
```

@ ELT MRAS,1,710612, 63737

```
000001          SUBROUTINE MRAS(CASE)                                    20000
000002          IF NMM(CASE) EQ 0, GO TO 50                              20010
000003          LET STOS = 0.                                            20020
000004          LET XTOS = 0.                                            20030
000005          DO, FOR EACH NOTIF IN NSET(CASE)                         20040
000006          IF OST(NOTIF) GR XTOS, LET XTOS = OST(NOTIF)             20050
000007          LET STOS = STOS + OST(NOTIF)                             20060
000008          LOOP                                                     20070
000009          LET TE = (1.0-GAMMA(CASE))*XTOS + GAMMA(CASE).STOS       20080
000010          DO, FOR EACH NOTIF IN NSET(CASE)                         20090
000011          CAUSE NOTIF AT DELTA(NOTIF)*(TE-OST(NOTIF)) + TIME       20100
000012          LET NUMBR(NOTIF) = 0                                     20110
000013          LET COMP(NOTIF) = 0                                      20120
000014          LOOP                                                     20130
000015       50 IF MMM(CASE) NE 0, CALL TOM(CASE)                        20140
000016          RETURN                                                   20150
000017          END OF MRAS                                              20160
```

```
@ ELT NOTIF,1,71-512, 837-(

000001          ENDOGENOUS EVENT NOTIF                                  35010
000002          LET JRAP = RAP                                          35030
000003          IF RAP EQ 4, LET JRAP = 5                               35040
000004          IF RAP EQ 5, LET JRAP = 6                               35050
000005        7 CALL CRES(CAS(NOTIF),NEED(NOTIF),*KONTR,*II)            35060
000006          IF(II)EQ(0), GO TO 33                                   35070
000007          IF (KONTR)EQ(0), GO TO 87                               35080
000008          IF KONTR GR 1, GO TO 15
000009          DO TO 10, FOR EACH IRS OF RQUE
000010          IF IB(IRS) LE 1, GO TO 10
000011          IF MFLG(IFLT(IRS)) NE 9, GO TO 10
000012          IF ACASE(IFLT(IRS)) NE CAS(NOTIF), GO TO 10
000013          LET KRES(FITON(IFLT(IRS))) = 0
000014          LET TVEC(IRS) = 0.0
000015          LET IRES = IRS
000016          GO TO 18
000017       10 LOOP
000018          GO TO 15                                                35090
000019       33 LET FAIL1(STATN(CAS(NOTIF)))=FAIL1(STATN(CAS(NOTIF)))+1 35100
000020          LET ITOL(CAS(NOTIF)) = 3                                35110
000021          GO TO 89                                                35120
000022       87 LET FAIL2(STATN(CAS(NOTIF)))=FAIL2(STATN(CAS(NOTIF)))+1 35130
000023          LET ITOL(CAS(NOTIF)) = 4                                35140
000024       89 LET COMP(NOTIF) = 4                                     35150
000025          CALL WRECK(NOTIF)                                       35160
000026          RETURN                                                  35170
000027       15 CALL VEC(CAS(NOTIF))                                    35180
000028          CALL OSET(CAS(NOTIF),KONTR)                             35190
000029          CALL RESAP(CAS(NOTIF),JRAP,*IRES)                       35200
000030       18 LET KRES(NOTIF) = IRES
000031          IF(IRES)EQ(0), GO TO 11                                 35220
000032          LET COMP(NOTIF) = 2                                     35230
000033          CALL SERVE(CAS(NOTIF),NOTIF,IRES)                       35240
000034          RETURN                                                  35250
000035       11 LET M=0                                                 35260
000036          CALL QUEUE(NOTIF,M)                                     35270
000037          RETURN                                                  35280
000038          END OF NOTIF                                            35290
```

```
000001          SUBROUTINE TOW(CASE)                                    25000
000002          CALL DTD(CASE,*XD,*YD,*D)                               25010
000003          LET GAP = D/2.0                                         25012
000004          IF HO LS GAP, LET GAP = HO                              25013
000005          LET M = NMM(CASE)                                       25020
000006          DO TO 50, FOR I=(1) (M)                                 25030
000007          CREATE NOTIF                                            25040
000008          FILE NOTIF IN NSET(CASE)                                25042
000009          LET NEED(NOTIF) = ITOW(CASE)                            25050
000010          LET PRI(NOTIF) = PRI(CASE)                              25060
000011          LET CAS(NOTIF) = CASE                                   25070
000012          LET NUMBR(NOTIF) = I                                    25080
000013          LET RESA(NOTIF) = 0                                     25090
000014          LET KRES(NOTIF) = 0                                     25100
000015          LET FLG(NOTIF) = 2                                      25110
000016          LET COMP(NOTIF) = 5                                     25120
000017          LET SIGNL(NOTIF) = 2
000018          IF I NE M, GO TO 40                                     25130
000019       60 LET XHAND(NOTIF) = XD
000020          LET YHAND(NOTIF) = YD                                   25150
000021          GO TO 45                                                25160
000022       40 IF D LE 0.0, GO TO 60
000023          LET YHAND(NOTIF) = YD + ,GAP*(YC(CASE)-YD))/D
000024          LET XHAND(NOTIF) = XD + (GAP*(XC(CASE)-XD))/D           25180
000025       45 IF NNN(CASE) GR 0, GO TO 50                             25190
000026          IF I EQ 2, GO TO 50                                     25200
000027          LET COMP(NOTIF) = 0                                     25210
000028          CAUSE NOTIF AT TIME                                     25220
000029       50 LOOP                                                    25230
000030          RETURN                                                  25240
000031          END OF TOW                                              25250
```

```
@ ELT DTD,1,710519, 63780

000001          SUBROUTINE DTD(CASE,XD,YD,D)                                30000
000002          IF NPATRL LE 0, GO TO 10                                    30010
000003          DO TO 5, FOR I = (1)(NPATRL)                                30020
000004          IF STATN(CASE) NE IPAT(I), GO TO 5                          30030
000005          LET XD = XS(ADJS(STATN(CASE),1))                            30040
000006          LET YD = YS(ADJS(STATN(CASE),1))                            30050
000007          GO TO 3                                                     30060
000008        5 LOOP                                                        30070
000009       10 LET XD = XS(STATN(CASE))                                    30080
000010          LET YD = YS(STATN(CASE))                                    30090
000011        3 LET D = SQRT((XC(CASE)-XD)**2 + (YC(CASE)-YD)**2)           30240
000012          RETURN                                                      30250
000013          END OF DTD                                                  30260
```

```
@ ELT SRCH,1,710E19, 63740

                SUBROUTINE SRCH(CASE)
        C
        C       SUBROUTINE SRCH CREATES AN EVENT NOTICE ENTITY (NOT.) FOR EACH
        C       SEARCH NEED.  SRCH THEN ATTEMPTS TO ASSIGN THE FIRST RESOURCE TO
        C       COMPLETE SM(I).
        C
                DO TO 10, FOR I=(1)(SIS,CASE))
                CREATE NOTE
                LET SM(NOTE) = PRTSM(SIS(CASE),I)*TSM(CASE)
                LET PRI(NOTE) = PRI(CASE)
                LET FLG(NOTE) = 3
                LET SDAY(NOTE) = 1
                LET SASG(NOTE) = 0
                LET ESAC(NOTE) = CASE
                LET SFLAG(NOTE) = 0
                LET SIGNL(NOTE) = 2
                LET RESA(NOTE) = 0
                LET RSRC(NOTE) = 0
                FILE NOTE IN SRHS(CASE)
             10 LOOP
                LET X = AINT(TIME)
                LET SN = X + RISE
                IF (TIME)LS(SN-XRX), GO TO 100
                LET SN = X + SET
                IF (TIME)LS(SN), GO TO 500
            100 LET K = 0
                DO TO 190, FOR EACH I IN SRHS(CASE)
                IF (K)NE(0), GO TO 150
                CALL SASS(CASE,SM(I),I,*IJK)
                IF (IJK)NE(0), GO TO 300
                LET JRAP = RAP
                IF (RAP)EQ(4), LET JRAP = 2
                IF (RAP)EQ(5), LET JRAP = 6
                CALL RESAP(CASE,JRAP,*IR)
                IF (IR)EQ(0), GO TO 110
                LET SFLAG(I) = 1
                LET RSRC(I) = IR
                LET SASG(I) = 1
                CALL SSS(I,IR,1)
                GO TO 120
            110 FILE I IN LIST
            120 LET K = K + 1
                GO TO 190
            150 FILE I IN LIST
            190 LOOP
                RETURN
            200 LET K = 0
                DO TO 290, FOR EACH I IN SRHS(CASE)
                IF (K)NE(0), GO TO 250
                CALL SASS(CASE,SM(I),I,*IJK)
                IF (IJK)NE(0), GO TO 300
                LET JRAP = RAP
                IF (RAP)EQ(4), LET JRAP = 2
                IF (RAP)EQ(5), LET JRAP = 6
```

```
000055          CALL RESAP(CASE,JRAP,*I-)
000056          IF (IR)EQ(0), GO TO 216
000057          LET SFLAG(I) = 1
000058          LET SASG(I) = 1
000059          LET RSRC(I) = IR
000060          CALL SSS(I,IR,1)
000061          GO TO 220
000062      216 LET SIGNL(I) = 0
000063          LET TINO(ESAC(I)) = TIME
000064          LET NQUE(ESAC(I)) = NQUE(ESAC(I)) + 1
000065          LET REA(ESAC(I)) = 1
000066          FILE I IN CQUE
000067      220 LET K = K + 1
000068          GO TO 290
000069      250 CAUSE NOTE CALLED I AT TIME + EPSLN
000070      290 LOOP
000071          RETURN
000072      300 REMOVE FIRST NOTE FROM SRHS(CASE)
000073          DESTROY NOTE
000074          IF SRHS(CASE) IS NOT EMPTY, GO TO 300
000075          IF (IJK)EQ(1), GO TO 310
000076          IF (IJK)EQ(2), GO TO 320
000077          IF (IJK)EQ(3), GO TO 320
000078      310 LET ITOL(CASE) = 3
000079          LET FAIL1(STATN(CASE)) = FAIL1(STATN(CASE)) + 1
000080          GO TO 330
000081      320 LET ITOL(CASE) = 4
000082          LET FAIL2(STATN(CASE)) = FAIL2(STATN(CASE)) + 1
000083      330 FILE CASE IN EXCS
000084          RETURN
000085          END
```

```
000001                ENDOGENOUS EVENT NOTE
000002          C
000003          C     ENDOGENOUS EVENT NOTE OCCURS AT TIME+EPSLN FOR SEARCH NEEDS,
000004          C     SM(I) WHERE I=(2)(S1); NOTE IS CAUSED ONLY IF CASE OCCURS DURING
000005          C     THE DAY
000006          C
000007                LET X = AINT (TIME)
000008                LET SN = X + RISE
000009                IF (TIME)LS(SN-XRX), GO TO 500
000010                LET SN = X + SET
000011                IF (TIME)GE(SN), GO TO 500
000012                CALL SASS(ESAC(NOTE),SM(NOTE),NOTE,*IJK)
000013                LET JRAP = RAP
000014                IF (RAP)EQ(4), LET JRAP = 2
000015                IF (RAP)EQ(5), LET JRAP = 6
000016                CALL RESAP(ESAC(NOTE),JRAP,*IR)
000017                IF (IR)EQ(0), GO TO 300
000018                LET SASG(NOTE) = 1
000019                LET RSRC(NOTE) = IR
000020                LET I = 0
000021                DO TO 200, FOR EACH LL IN SRHS(ESAC(NOTE))
000022                IF (SFLAG(LL))EQ(1), LET I = 1
000023           200  LOOP
000024                IF (I)EQ(0), LET SFLAG(NOTE) = 1
000025                IF (SFLAG(NOTE))EQ(1), GO TO 210
000026                CALL SSS(NOTE,IR,2)
000027                GO TO 220
000028           210  CALL SSS(NOTE,IR,1)
000029           220  RETURN
000030           300  LET I = 0
000031                DO TO 305, FOR EACH LL IN SRHS(ESAC(NOTE))
000032                IF (SIGNL(LL))NE(2), LET I = 1
000033           305  LOOP
000034                IF (I)EQ(0), LET TINQ(ESAC(NOTE)) = TIME
000035                LET NQUE(ESAC(NOTE)) = NQUE(ESAC(NOTE)) + 1
000036                IF (REA(ESAC(NOTE)))EQ(2), LET REA(ESAC(NOTE)) = 1
000037                LET SIGNL(NOTE) = 0
000038                FILE NOTE IN CQUE
000039                RETURN
000040           500  FILE NOTE IN LIST
000041                RETURN
000042                END
```

```
@ ELT SASS,1,710619, A3730

                SUBROUTINE SASS(CASE,X,IRS,*OTH,IDC)
          C
          C     SUBROUTINE SASS MAY BE CALLED FROM THE SERVICE ROUTINE WHEN
          C     S(2)=2 OR FROM THE SEARCH ROUTINE.  IT FINDS AN ORDERED SET OF
          C     RESOURCES TO SERVE THE SEARCH CASE AND PLACES THEM IN RQUE.
          C
                LET IDC = 0
                CALL CRES(CASE,IB,*KNT,*IJ)
                IF (IJ)NE(0), GO TO 5
                LET IDC = 1
                RETURN
              5 IF (KNT)NE(0), GO TO 6
                LET IDC = 2
                RETURN
              6 CALL VEC(CASE)
                LET SNS = AINT(TIME) + SET
                IF (TIME)GE(SNS), LET SNS = SNS + 1.0
                IF (TIME+TOLS(PRI(CASE)))GR(SNS), GO TO 7
                LET TOLRS = TOLS(PRI(CASE))
                GO TO 8
              7 LET TOLRS = SNS - TIME
          C
          C     CALCULATE PR FOR EACH RESOURCE IN RQUE
          C
              8 DO TO 9, FOR EACH IRS IN RQUE
                LET X = XS(STN(IRS))-XC(CASE)
                LET Y = YS(STN(IRS))-YC(CASE)
                LET D = SQRT(X*X+Y*Y)
                IF (SWELL(CASE))GR(SLIM(TYPE(IRS))),GO TO 100
                LET X = D/SOA1(TYPE(IRS))
                GO TO 101
            100 LET X = D/SOA2(TYPE(IRS))
            101 LET Y = END(TYPE(IRS))-(2.0*X)
                IF (Y)LE(0.0), GO TO 150
                LET D = END(TYPE(IRS)) + TF(TYPE(IRS))
                LET STOLR = TOLRS
                IF (VOP(TYPE(IRS)))EQ(0), LET STOLR = TOLS(PRI(CASE))
                LET SNS = AINT(STOLR/D)
                LET ANS = STOLR - (D*SNS) - X
                IF (ANS)GE(0.0), GO TO 102
                LET ANS = 0.0
                GO TO 103
            102 IF (ANS)LE(Y), GO TO 103
                LET ANS = Y
            103 LET ANS = (Y*SNS) + ANS
                IF (ANS)LS(0.0), GO TO 150
                IF (XMLES)LE(0.0), GO TO 150
                LET PR(IRS) = (SOA3(TYPE(IRS))/XMLES)*ANS
                GO TO 9
            150 LET PR(IRS) = 0.0
              9 LOOP
                DO, FOR EACH IRS IN RQUE
                LET TST = XMLES/SOA3(TYPE(IRS))
                LET COST(IRS) = COSTD(TYPE(IRS))*(TVEC(IRS)+TST)
```

```
000055                LET DVEC(IRS) = TVEC(IRS)
000056                IF (EIAT(IRS))EQ(0.0), LET DVEC(IRS)=TVEC(IRS)+DLAY(TYPE(IRS))
000057                LOOP
000058            999 IF (S2S(CASE))EQ(2), GO TO 20
000059                IF (SDAY(NOTE))EQ(1), LET PERC = PDC1
000060                IF (SDAY(NOTE))GR(1), LET PERC = PDC2
000061                GO TO 21
000062             20 LET PERC = 1.0
000063             21 IF RQUE IS EMPTY, GO TO 31
000064                REMOVE FIRST IRS FROM RQUE
000065                IF (PR(IRS))GR(0.0), GO TO 40
000066                LET X = XS(STN(IRS))-XC(CASE)
000067                LET Y = YS(STN(IRS))-YC(CASE)
000068                LET D = SQRT(X*X+Y*Y)
000069                IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 25
000070                LET X = D/SDA1(TYPE(IRS))
000071                GO TO 26
000072             25 LET X = D/SDA2(TYPE(IRS))
000073             26 IF (END(TYPE(IRS))-(2.0*X))LE(0.0), GO TO 21
000074                FILE IRS IN TSET
000075                GO TO 21
000076             40 IF (PR(IRS))GE(PERC), GO TO 50
000077                FILE IRS IN PSET
000078                GO TO 21
000079             50 FILE IRS IN CSET
000080             30 GO TO 21
000081             31 IF CSET IS EMPTY, GO TO 60
000082                REMOVE FIRST IRS FROM CSET
000083                FILE IRS IN RQUE
000084                GO TO 31
000085             60 IF PSET IS EMPTY, GO TO 70
000086                REMOVE FIRST IRS FROM PSET
000087                FILE IRS IN RQUE
000088                GO TO 60
000089             70 IF TSET IS EMPTY, GO TO 80
000090                REMOVE FIRST IRS FROM TSET
000091                FILE IRS IN RQUE
000092                GO TO 70
000093             80 IF RQUE IS EMPTY, LET IDC = 3
000094                RETURN
000095                END
```

```
000001                SUBROUTINE SSS(NOTE,IRES,KLK)
000002          C
000003          C     SSS IS THE SEARCH SERVICE ROUTINE, IT CAN NOT BE CALLED FROM EXQ'
000004          C
000005                LET IVAR = 0
000006                IF (KLK)EQ(1), GO TO 200
000007                IF (KLK)EQ(3), GO TO 200
000008                IF (VOP(TYPE(IRES)))EQ(0), GO TO 200
000009             10 LET SN = AINT(TIME)
000010                LET SN = SN + SET
000011                LET SN = SN-TIME
000012                LET DVEC(IRES) = TVEC(IRES)
000013                IF (EIAT(IRES))EQ(0.0),LET DVEC(IRES)=TVEC(IRES)+DLAY(TYPE(IRES))
000014                IF (DVEC(IRES))LS(SN), GO TO 200
000015                LET SN = AINT(TIME)
000016                LET SN = SN + RISE + 1.0
000017                IF (TIME+DVEC(IRES))GE(SN), GO TO 200
000018                FILE NOTE IN LIST
000019                LET RSRC(NOTE) = 0
000020                IF (EIAT(IRES))NE(0.0), GO TO 11
000021                LET BCREW(STN(IRES)) = BCREW(STN(IRES)) - 1
000022                RETURN
000023             11 IF (IB(IRES))EQ(0), GO TO 50
000024                IF (IB(IRES))EQ(1), GO TO 40
000025          C
000026          C     IF IB(IRES)=2 AND RESOURCE WILL NOT BE USED:
000027          C
000028                LET IB(IRES) = 0
000029                GO TO (30,31,30,30,30,30,30,30,30,30,30,30,32),MFLG(IFLT(IRES))
000030             31 CANCEL ONSCN CALLED IDEV(IFLT(IRES))
000031                DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000032                GO TO 30
000033             32 CANCEL CHEKN CALLED IDEV(IFLT(IRES))
000034                DESTROY CHEKN CALLED IDEV(IFLT(IRES))
000035             30 LET MFLG(IFLT(IRES)) = 9
000036                CALL EXQ(IRES,**IVAR)
000037                RETURN
000038             40 IF (DEP(IFLT(IRES)))NE(0.0), GO TO 45
000039          C
000040          C     IF IB(IRES)=1, DEST = 0.0 AND RESOURCE WILL NOT BE USED:
000041          C
000042                GO TO (64,60,64,61,62,63,64,64,64,64,65,66,64),MFLG(IFLT(IRES))
000043             60 CANCEL ONSCN CALLED IDEV(IFLT(IRES))
000044                DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000045                GO TO 41
000046             61 CANCEL COMPL CALLED IDEV(IFLT(IRES))
000047                DESTROY COMPL CALLED IDEV(IFLT(IRES))
000048                GO TO 41
000049             62 CANCEL SSET CALLED IDEV(IFLT(IRES))
000050                DESTROY SSET CALLED IDEV(IFLT(IRES))
000051                GO TO 41
000052             63 CANCEL FUEL CALLED IDEV(IFLT(IRES))
000053                DESTROY FUEL CALLED IDEV(IFLT(IRES))
000054                GO TO 41
```

```
000055          45  CANCEL DELAY CALLED IDEV(IFLT(IRES))
000056              DESTROY DELAY CALLED IDEV(IFLT(IRES))
000057              GO TO 41
000058          66  CANCEL READY CALLED IDEV(IFLT(IRES))
000059              DESTROY READY CALLED IDEV(IFLT(IRES))
000060          41  LET IB(IRES) = 0
000061              LET MFLG(IFLT(IRES)) = 0
000062              CALL EXQ(IRES,**IVAR)
000063              RETURN
000064          64  WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000065              FORMAT (' RESOURCE ',2I4,' WENT HAYWIRE AT STATEMENT 40 IN SSS')
000066              RETURN
000067        C
000068        C     IF IB(IRES)=1, DEST NE 0.0 AND RESOURCE WILL NOT BE USED.
000069        C
000070          45  GO TO (70,73,72,71,71,71,71,71,71,71,71,71,71),MFLG(IFLT(IRES))
000071          70  CANCEL ARVSN CALLED IDEV(IFLT(IRES))
000072              DESTROY ARVSN CALLED IDEV(IFLT(IRES))
000073              GO TO 46
000074          72  CANCEL ARSCH CALLED IDEV(IFLT(IRES))
000075              DESTROY ARSCH CALLED IDEV(IFLT(IRES))
000076              GO TO 46
000077          73  CANCEL ONSCN CALLED IDEV(IFLT(IRES))
000078              DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000079          46  LET IB(IRES) = 0
000080              LET XR(IRES) = XI(IFLT(IRES))
000081              LET YR(IRES) = YI(IFLT(IRES))
000082              LET XDEST(IFLT(IRES)) = 0.0
000083              LET YDEST(IFLT(IRES)) = 0.0
000084              LET MFLG(IFLT(IRES)) = 0
000085              CALL EXQ(IRES,**IVAR)
000086          50  RETURN
000087          71  WRITE ON TAPE 6, IRES, MFLG(IFLT(IRES))
000088              FORMAT (' RESOURCE ',2I4,' WENT HAYWIRE AT STATEMENT 45 IN SSS ',
000089              RETURN
000090        C
000091        C     UPDATING ALL NECESSARY RES/FLT ATTRIBUTES.
000092        C
000093         200  IF (EIAT(IRES))NE(0.0), GO TO 300
000094              CREATE FLT CALLED IFLT(IRES)
000095              LET EIAT(IRES) = 1.0
000096              LET TFLT(IFLT(IRES)) = 0.0
000097              LET TOW(IFLT(IRES)) = 0
000098              LET HCREW(IFLT(IRES)) = 0
000099              LET IB(IRES) = 1
000100              LET NCASE(IRES) = NCASE(IRES) + 1
000101              LET PRIOR(IRES) = PRI(NOTE)
000102              LET CNRES(ESAC(NOTE)) = CNRES(ESAC(NOTE)) + 1
000103              LET COSTC(ESAC(NOTE)) = COSTC(ESAC(NOTE)) + COST(IRES)
000104              LET ACASE(IFLT(IRES)) = ESAC(NOTE)
000105              LET DEP(IFLT(IRES)) = 0.0
000106              LET FITON(IFLT(IRES)) = NOTE
000107              LET XDEST(IFLT(IRES)) = 0.0
000108              LET YDEST(IFLT(IRES)) = 0.0
000109              CREATE READY CALLED IDEV(IFLT(IRES))
000110              LET NREAD(IDEV(IFLT(IRES))) = NOTE
000111              LET RREAD(IDEV(IFLT(IRES))) = IRES
000112              LET FREAD(IDEV(IFLT(IRES))) = KLK
```

```
000113          LET MFLG(IFLT(IRES)) = 12
000114          CAUSE READY CALLED IDEV(IFLT(IRES)) AT TIME+DELAY(TYPE(IRES))
000115          LET NEEDS(STN(IRES)) = NEEDS(STN(IRES)) + 1
000116          RETURN
000117      300 IF (IB(IRES))NE(0), GO TO 400
000118          IF (MFLG(IFLT(IRES)))EQ(9), GO TO 301
000119          IF (MFLG(IFLT(IRES)))NE( 7), GO TO 80
000120          CANCEL  HOME CALLED IDEV(IFLT(IRES))
000121          DESTROY HOME CALLED IDEV(IFLT(IRES))
000122          GO TO 301
000123       80 WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000124          FORMAT (' RESOURCE ',2I4,' WENT HAYWIRE AT STATEMENT 300 IN SSS')
000125      301 LET XR(IRES) = XI(IFLT(IRES))
000126          LET YR(IRES) = XI(IFLT(IRES))
000127          GO TO 600
000128      400 IF (DEP(IFLT(IRES)))NE(0.0), GO TO 500
000129          IF (MFLG(IFLT(IRES)))EQ(9), GO TO 401
000130          IF (MFLG(IFLT(IRES)))NE( 2), GO TO 90
000131          CANCEL  ONSCN CALLED IDEV(IFLT(IRES))
000132          DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000133          GO TO 401
000134       90 IF (MFLG(IFLT(IRES)))NE( 4), GO TO 91
000135          CANCEL  COMPL CALLED IDEV(IFLT(IRES))
000136          DESTROY COMPL CALLED IDEV(IFLT(IRES))
000137          GO TO 401
000138       91 IF (MFLG(IFLT(IRES)))NE( 5), GO TO 92
000139          CANCEL  SSET CALLED IDEV(IFLT(IRES))
000140          DESTROY SSET CALLED IDEV(IFLT(IRES))
000141          GO TO 401
000142       92 IF (MFLG(IFLT(IRES)))NE( 6), GO TO 93
000143          CANCEL  FUEL CALLED IDEV(IFLT(IRES))
000144          DESTROY FUEL CALLED IDEV(IFLT(IRES))
000145          GO TO 401
000146       93 IF (MFLG(IFLT(IRES)))NE(11), GO TO 94
000147          CANCEL  DELAY CALLED IDEV(IFLT(IRES))
000148          DESTROY DELAY CALLED IDEV(IFLT(IRES))
000149          GO TO 401
000150       94 IF (MFLG(IFLT(IRES)))NE(12), GO TO 95
000151          CANCEL READY CALLED IDEV(IFLT(IRES))
000152          DESTROY READY CALLED IDEV(IFLT(IRES))
000153          GO TO 401
000154       95 IF (MFLG(IFLT(IRES)))NE(13), GO TO 96
000155          CANCEL  CHEKN CALLED IDEV(IFLT(IRES))
000156          DESTROY CHEKN CALLED IDEV(IFLT(IRES))
000157          GO TO 401
000158       96 WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000159          FORMAT (' RESOURCE ',2I4,' WENT HAYWIRE AT STATEMENT 400 IN SSS')
000160      401 IF IWAIT(ACASE(IFLT(IRES))) EQ IRES, LET IWAIT(ACASE(IFLT(IRES)))
000161        * = 0
000162          GO TO 600
000163      500 IF (MFLG(IFLT(IRES)))EQ(9), GO TO 501
000164          IF (MFLG(IFLT(IRES)))NE( 1), GO TO 100
000165          CANCEL  ARVSN CALLED IDEV(IFLT(IRES))
000166          DESTROY ARVSN CALLED IDEV(IFLT(IRES))
000167          GO TO 501
000168      100 IF (MFLG(IFLT(IRES)))NE( 3), GO TO 101
000169          CANCEL  ARSCH CALLED IDEV(IFLT(IRES))
000170          DESTROY ARSCH CALLED IDEV(IFLT(IRES))
```

```
000171                GO TO 501
000172          101   IF (MFLG(IFLT(IRES)))NE(2), GO TO 102
000173                CANCEL ONSCN CALLED IDEV(IFLT(IRES))
000174                DESTROY ONSCN CALLED IDEV(IFLT(IRES))
000175                GO TO 501
000176          102   WRITE ON TAPE 6, IRES,MFLG(IFLT(IRES))
000177                FORMAT (' RESOURCE ',2I4,' MENT HAYWIRE AT STATEMENT 500 IN SSS')
000178          501   LET XR(IRES) = XI(IFLT(IRES))
000179                LET YR(IRES) = YI(IFLT(IRES))
000180          600   LET IB(IRES) = 1
000181                LET NCASE(IRES) = NCASE(IRES) + 1
000182                LET PRIOR(IRES) = PRI(NOTE)
000183                LET CNRES(ESAC(NOTE)) = CNRES(ESAC(NOTE)) + 1
000184                LET COSTC(ESAC(NOTE)) = COSTC(ESAC(NOTE)) + COST(IRES)
000185                LET ACASE(IFLT(IRES)) = ESAC(NOTE)
000186                LET DEP(IFLT(IRES)) = TIME
000187                LET FITON(IFLT(IRES)) = NOTE
000188                LET XDEST(IFLT(IRES)) = XC(ESAC(NOTE))
000189                LET YDEST(IFLT(IRES)) = YC(ESAC(NOTE))
000190        C
000191        C       CREATE AND CAUSE ARSCH FOR SEARCH NEED.
000192        C
000193                CREATE ARSCH CALLED IDEV(IFLT(IRES))
000194                LET LNOTE(IDEV(IFLT(IRES))) = NOTE
000195                LET LRES(IDEV(IFLT(IRES))) = IRES
000196                LET LFLG(IDEV(IFLT(IRES))) = KLK
000197                LET MFLG(IFLT(IRES)) = 3
000198                LET ROS(IFLT(IRES)) = TIME + TVEC(IRES)
000199                CAUSE ARSCH CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000200                LET NEEDS(STN(IRES)) = NEEDS(STN(IRES)) + 1
000201          999   RETURN
000202                END
```

```
000001                ENDOGENOUS EVENT READY
000002          C
000003          C     ENDOGENOUS EVENT READY OCCURS AT THE END OF THE DELAY TIME
000004          C     NECESSARY TO READY A RESOURCE AT IT'S HOME STATION BEFORE SERVING
000005          C     A LONG SEARCH NEED.
000006          C
000007                STORE NREAD(READY) IN NOTE
000008                STORE RREAD(READY) IN IRS
000009                STORE FREAD(READY) IN LFG
000010                DESTROY READY
000011                LET X = XC(ESAC(NOTE)) - XR(IRS)
000012                LET Y = YC(ESAC(NOTE)) - YR(IRS)
000013                LET D = SQRT(X*X+Y*Y)
000014                IF (SMELL(ESAC(NOTE)))GR(SLIM(TYPE(IRS))), GO TO 10
000015                LET TVEC(IRS) = D/SOA1(TYPE(IRS))
000016                GO TO 20
000017             10 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
000018             20 LET DEP(IFLT(IRS)) = TIME
000019                LET TFLT(IFLT(IRS)) = TIME
000020                LET MFLG(IFLT(IRS)) = 3
000021                LET ROS(IFLT(IRS)) = TIME + TVEC(IRS)
000022                LET XDEST(IFLT(IRS)) = XC(ESAC(NOTE))
000023                LET YDEST(IFLT(IRS)) = YC(ESAC(NOTE))
000024                CREATE ARSCH CALLED IDEV(IFLT(IRS))
000025                LET LNOTE(IDEV(IFLT(IRS))) = NOTE
000026                LET LRES(IDEV(IFLT(IRS))) = IRS
000027                LET LFLG(IDEV(IFLT(IRS))) = LFG
000028                CAUSE ARSCH CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
000029                RETURN
000030                END
```

```
000001                ENDOGENOUS EVENT ARSCH
000002          C
000003          C     ARSCH OCCURS WHEN A RESOURCE ARRIVES ON SCENE TO SERVE A SEARCH
000004          C     NEED.
000005          C
000006                STORE LNOTE(ARSCH) IN NOTE
000007                STORE LRES(ARSCH) IN IRS
000008                LET LFG = LFLG(ARSCH)
000009                DESTROY ARSCH
000010                IF (RESA(ESAC(NOTE)))EQ(0), LET RESA(ESAC(NOTE)) = IRS
000011                IF (RESA(NOTE))EQ(0), LET RESA(NOTE) = IRS
000012                IF (ITOL(ESAC(NOTE)))NE(2), GO TO 201
000013                LET X = 0.0
000014                LET I = 0
000015                DO, FOR EACH JJ IN SRHS(ESAC(NOTE))
000016                IF (SIGNL(JJ))NE(2), LET I = 1
000017                LOOP
000018                IF (I)EQ(1), LET X = TIME - TINQ(ESAC(NOTE))
000019                LET TQUE1(ESAC(NOTE)) = TQUE(ESAC(NOTE)) + X
000020                LET NCAS(STN(IRS)) = NCAS(STN(IRS)) + 1
000021                LET TWAIT(ESAC(NOTE)) = TIME - OCCUR(ESAC(NOTE))
000022                LET AVGTW(STN(IRS)) = AVGTW(STN(IRS)) + TWAIT(ESAC(NOTE))
000023                CALL STATS(ESAC(NOTE),IRS)
000024                IF (TWAIT(ESAC(NOTE)))GR(TOL(PRI(ESAC(NOTE)))), GO TO 10
000025                LET ITOL(ESAC(NOTE)) = 1
000026                GO TO 201
000027             10 LET ITOL(ESAC(NOTE)) = 0
000028                LET FAIL3(STN(IRS)) = FAIL3(STN(IRS)) + 1
000029            201 LET XR(IRS) = XC(ESAC(NOTE))
000030                LET YR(IRS) = YC(ESAC(NOTE))
000031                LET XDEST(IFLT(IRS)) = 0.0
000032                LET YDEST(IFLT(IRS)) = 0.0
000033                LET DEP(IFLT(IRS)) = 0.0
000034                LET ROS(IFLT(IRS)) = TIME
000035                LET XVEC = XS(STN(IRS)) - XC(ESAC(NOTE))
000036                LET YVEC = YS(STN(IRS)) - YC(ESAC(NOTE))
000037                LET VECTR = SQRT(XVEC*XVEC+YVEC*YVEC)
000038                IF (SWELL(ESAC(NOTE)))GR(SLIM(TYPE(IRS))), GO TO 60
000039                LET VECTR = VECTR/SOA1(TYPE(IRS))
000040                GO TO 61
000041             60 LET VECTR = VECTR/SOA2(TYPE(IRS))
000042             61 LET X = END(TYPE(IRS))-(2.0*VECTR)
000043                IF (X)GE(SM(NOTE)/SOA3(TYPE(IRS))), GO TO 300
000044                LET TS(NOTE) = X
000045                IF (VOP(TYPE(IRS)))EQ(0), GO TO 100
000046                LET  X = AINT(TIME)
000047                LET  X = X + SET
000048                IF (X-TIME)LE(0.0), LET X = X + 1.0
000049                IF (TS(NOTE))GR(X-TIME), GO TO 400
000050            100 CREATE FUEL CALLED IDEV(IFLT(IRS))
000051                LET NFUEL(IDEV(IFLT(IRS))) = NOTE
000052                LET RFUEL(IDEV(IFLT(IRS))) = IRS
000053                LET MFLG(IFLT(IRS)) = 6
000054                CAUSE FUEL CALLED IDEV(IFLT(IRS)) AT TIME + TS(NOTE)
```

```
000055                RETURN
000056          350   LET TS(NOTE) = SM(NOTE)/SOA3(TYPE(IRS))
000057                IF (LFG)EQ(1), GO TO 350
000058                IF (VOP(TYPE(IRS)))EQ(0), GO TO 350
000059                LET X = AINT(TIME)
000060                LET X = X + SET
000061                IF (X-TIME)LE(0.0), LET X = X + 1.0
000062                IF (TS(NOTE))GR(X-TIME), GO TO 400
000063          350   CREATE COMPL CALLED IDEV(IFLT(IRS))
000064                LET MFLG(IFLT(IRS)) = 4
000065                LET NCMPL(IDEV(IFLT(IRS))) = NOTE
000066                LET RCMPL(IDEV(IFLT(IRS))) = IRS
000067                CAUSE COMPL CALLED IDEV(IFLT(IRS)) AT TIME + TS(NOTE)
000068                RETURN
000069          400   CREATE SSET CALLED IDEV(IFLT(IRS))
000070                LET TS(NOTE) = X - TIME
000071                LET MFLG(IFLT(IRS)) = 5
000072                LET ETON(IDEV(IFLT(IRS)),) = NOTE
000073                LET SER(IDEV(IFLT(IRS))) = IRS
000074                CAUSE SSET CALLED IDEV(IFLT(IRS)) AT X + 1.0/1440.0
000075                RETURN
000076                END
```

@ ELT COMPL,1,71519, 63781

```
000001                ENDOGENOUS EVENT COMPL
000002          C
000003          C     COMPL OCCURS WHEN A LONG SEARCH NEED HAS BEEN COMPLETED.
000004          C
000005                STORE NCMPL(COMPL) IN NOTE
000006                STORE RCMPL(COMPL) IN IRS
000007                DESTROY COMPL
000008                LET RSRC(NOTE) = 0
000009                LET SIS(ESAC(NOTE)) = SIS(ESAC(NOTE)) - 1
000010                LET IB(IRS) = 0
000011                LET MFLG(IFLT(IRS)) = 9
000012                LET IVAR = 0
000013                CALL EXQ(IRS,**IVAR)
000014                IF (SIS(ESAC(NOTE)))EQ(0), GO TO 200
000015                RETURN
000016            200 LET SIS(ESAC(NOTE)) = -1
000017                LET KJI = NNN(ESAC(NOTE)) + MMM(ESAC(NOTE))
000018                IF (KJI)EQ(0), GO TO 400
000019                IF (KJI)GR(1), GO TO 300
000020                CALL SRAS(ESAC(NOTE))
000021                RETURN
000022            300 CALL MRAS(ESAC(NOTE))
000023                RETURN
000024            400 CALL TERM(ESAC(NOTE))
000025                RETURN
000026                END
```

```
@ ELT SSET,I,710619, 63781

                    ENDOGENOUS EVENT SSET
          C
          C         SSET OCCURS AT SUNSET. SM(NOTE) IS UPDATED; NOTE IS PLACED IN
          C         THE SUNRISE LIST AND SUBROUTINE EXQ IS CALLED FOR THE IDLE
          C         RESOURCE.
          C
                    STORE ETON(SSET) IN NOTE
                    STORE SER(SSET) IN IR
                    DESTROY SSET
                    LET MFLG(IFLT(IR)) = 9
                    LET RSRC(NOTE) = 0
                    LET SM(NOTE) = SM(NOTE) - (TS(NOTE)*SQA3(TYPE(IR)))
                    FILE NOTE IN LIST
                    LET IB(IR) = 0
                    LET IVAR = 0
                    CALL EXQ(IR,**IVAR)
                    RETURN
                    END
```

```
@ ELT FUEL,1,710519, 63731

                ENDOGENOUS EVENT FUEL
        C
        C       FUEL OCCURS WHEN A RESOURCE MUST RETURN HOME TO REFUEL. SM(NOTE)
        C       IS UPDATED AND THE ENDOGENOUS EVENT HOMEF IS CREATED AND CAUSED.
        C
                STORE NFUEL(FUEL) IN NOTE
                STORE RFUEL(FUEL) IN IRS
                DESTROY FUEL
                LET SM(NOTE) = SM(NOTE) - (TS(NOTE)*SOA3(TYPE(IRS)),
                LET X = XR(IRS) - XS(STN(IRS))
                LET Y = YR(IRS) - YS(STN(IRS))
                LET X = SQRT(X*X+Y*Y)
                IF (SWELL(ESAC(NOTE)))GR(SLIM(TYPE(IRS))), GO TO 20
                LET TVEC(IRS) = X/SOA1(TYPE(IRS))
                GO TO 30
            20  LET TVEC(IRS) = X/SOA2(TYPE(IRS))
            30  LET DEP(IFLT(IRS)) = TIME
                LET HCREW(IFLT(IRS)) = 2
                LET XDEST(IFLT(IRS)) = XS(STN(IRS))
                LET YDEST(IFLT(IRS)) = YS(STN(IRS))
                CREATE HOMEF CALLED IDEV(IFLT(IRS))
                LET NHOM(IDEV(IFLT(IRS))) = NOTE
                LET RHOM(IDEV(IFLT(IRS))) = IRS
                LET MFLG(IFLT(IRS)) = 8
                CAUSE HOMEF CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
                RETURN
                END
```

@ ELT HOMEF,1,71-519, 63781

```
000001                ENDOGENOUS EVENT HOMEF
000002          C
000003          C     HOMEF OCCURS WHEN A RESOURCE ARRIVES AT IT'S HOME STATION TO
000004          C     REFUEL. ALL ATTRIBUTES ARE THEN UPDATED.
000005          C
000006                STORE NHOM(HOMEF) IN NOTE
000007                STORE RHOM(HOMEF) IN IRS
000008                DESTROY HOMEF
000009                LET XR(IRS) = XS(STN(IRS))
000010                LET YR(IRS) = YS(STN(IRS))
000011                LET DEP(IFLT(IRS)) = 0.0
000012                LET XDEST(IFLT(IRS)) = 0.0
000013                LET YDEST(IFLT(IRS)) = 0.0
000014                CREATE SNDBK CALLED IDEV(IFLT(IRS))
000015                STORE NOTE IN NSDBK(IDEV(IFLT(IRS)))
000016                STORE IRS IN RSDBK(IDEV(IFLT(IRS)))
000017                LET MFLG(IFLT(IRS)) = 10
000018                CAUSE SNDBK CALLED IDEV(IFLT(IRS)) AT TIME + TF(TYPE(IRS))
000019                RETURN
000020                END
```

```
000001                ENDOGENOUS EVENT SNDBK
000002          C
000003          C     SNDBK OCCURS AT THE END OF REFUELING TIME OF A RESOURCE SERVING
000004          C     A LONG SEARCH NEED; SUBROUTINE SASS IS CALLED TO OBTAIN AN ORDERED
000005          C     SET OF RESOURCES; THE BEST IDLE RESOURCE IS CHOSEN TO SERVE THE
000006          C     SEARCH NEED.
000007          C
000008                STORE NSDBK(SNDBK) IN NOTE
000009                STORE RSDBK(SNDBK) IN IRS
000010                DESTROY SNDBK
000011                LET RSRC(NOTE) = 0
000012                LET IB(IRS) = 0
000013                LET EIAT(IRS) = 0.0
000014                LET TEMP = TFLT(IFLT(IRS))
000015                IF (TEMP)LS(TLAST), LET TEMP = TLAST
000016                LET TUTIL(IRS) = TUTIL(IRS) + TIME - TEMP
000017                DESTROY FLT CALLED IFLT(IRS)
000018                LET BCREW(STN(IRS)) = BCREW(STN(IRS)) - 1
000019                CALL SASS(ESAC(NOTE),SM(NOTE),NOTE,*KJI)
000020                DO TO 100, FOR EACH IR IN RQUE
000021                IF (IB(IR))GR(0), GO TO 100
000022                IF (EIAT(IR))EQ(0.0), GO TO 10
000023                IF (HCREW(IFLT(IR)))EQ(0), GO TO 20
000024                GO TO 100
000025             10 CALL ROCA(ESAC(NOTE),IR,*II)
000026                IF (II)EQ(1), GO TO 20
000027                GO TO 100
000028             20 LET DVEC(IR) = TVEC(IR)
000029                IF (EIAT(IR))EQ(0.0), LET DVEC(IR) = TVEC(IR) + DLAY(TYPE(IR))
000030                IF (IR)EQ(IRS), LET DVEC(IR) = TVEC(IR)
000031                LET SN = AINT(TIME) + RISE - XRX
000032                IF (TIME)GE(SN), GO TO 1
000033                IF (TIME+DVEC(IR))GE(AINT(TIME)+RISE), GO TO 200
000034                IF (EIAT(IR))EQ(0.0), LET BCREW(STN(IR))=BCREW(STN(IR))-1
000035                GO TO 100
000036              1 LET SN = AINT(TIME) + SET
000037                IF (TIME)GE(SN), GO TO 2
000038                IF (DVEC(IR))LS(SN-TIME), GO TO 200
000039                LET SN = AINT(TIME) + RISE + 1.0
000040                IF (TIME+DVEC(IR))GE(SN), GO TO 200
000041                IF (EIAT(IR))EQ(0.0), LET BCREW(STN(IR))=BCREW(STN(IR))-1
000042                GO TO 100
000043              2 LET SN = AINT(TIME) + RISE + 1.0
000044                IF (TIME+DVEC(IR))GE(SN), GO TO 200
000045                IF (EIAT(IR))EQ(0.0), LET BCREW(STN(IR))=BCREW(STN(IR))-1
000046                GO TO 100
000047            200 IF (IR)EQ(IRS), GO TO 300
000048                LET RSRC(NOTE) = IR
000049                LET SASG(NOTE) = 1
000050                CALL SSS(NOTE,IR,3)
000051                RETURN
000052            300 CREATE FLT CALLED IFLT(IRS)
000053                STORE ESAC(NOTE) IN ACASE(IFLT(IRS))
000054                LET DEP(IFLT(IRS)) = TIME
```

```
000055                STORE NOTE IN FITOP(IFLT(IRS))
000056                LET HCRF'(IFLT(IRS)) = F
000057                LET MFLG(IFLT(IRS)) = 3
000058                LET ROS(IFLT(IRS)) = TIME + TVEC(IRS)
000059                LET TFLT(IFLT(IRS)) = TIME
000060                LET TOM(IFLT(IRS)) = 0
000061                LET XDEST(IFLT(IRS)) = XC(ESAC(NOTE))
000062                LET YDEST(IFLT(IRS)) = YC(ESAC(NOTE))
000063                LET EIAT(IRS) = 1.0
000064                LET IB(IRS) = 1
000065                LET RSRC(NOTE) = IRS
000066                LET SASG(NOTE) = 1
000067                CREATE ARSCH CALLED IDEV(IFLT(IRS))
000068                STORE NOTE IN LNOTE(IDEV(IFLT(IRS)))
000069                STORE IRS IN LRES(IDEV(IFLT(IRS)))
000070                LET LFLG(IDEV(IFLT(IRS))) = 3
000071                CAUSE ARSCH CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
000072                RETURN
000073            100 LOOP
000074                FILE NOTE IN LIST
000075                RETURN
000076                END
```

```
@ ELT SRISE,1,71-519, 63791

                    ENDOGENOUS EVENT SRISE
          C
          C         SRISE OCCURS AT SUNRISE-XXX; THE SUNRISE LIST IS EXAMINED AND
          C         RESOURCES ARE VECTORED TO ARRIVE ON THE SEARCH SCENE AT SUNRISE
          C         OR SOON THEREAFTER.
          C
             11     IF LIST IS EMPTY, GO TO 999
                    REMOVE FIRST NOTE FROM LIST
                    LET I = 0
                    DO, FOR EACH II IN SRHS(ESAC(NOTE))
                    IF (SASG(II))EQ(1), LET I = 1
                    LOOP
                    IF (I)EQ(0), GO TO 10
                    DO, FOR EACH II IN SRHS(ESAC(NOTE))
                    LET SDAY(II) = SDAY(II) + 1
                    LET SASG(II) = 0
                    LOOP
             10     CALL SASS(ESAC(NOTE),SM(NOTE),NOTE,*IC)
                    LET JRAP = RAP
                    IF (RAP)EQ(4), LET JRAP = 2
                    IF (RAP)EQ(5), LET JRAP = 6
                    CALL RESAP(ESAC(NOTE),JRAP,*IRES)
                    IF (IRES)NE(0), GO TO 20
                    LET NQUE(ESAC(NOTE)) = NQUE(ESAC(NOTE)) + 1
                    IF (REA(ESAC(NOTE)))EQ(2), LET REA(ESAC(NOTE)) = 1
                    LET I = 0
                    DO, FOR EVERY JJ IN SRHS(ESAC(NOTE))
                    IF (SIGNL(JJ))NE(2), LET I = 1
                    LOOP
                    IF (I)EQ(0), LET TINQ(ESAC(NOTE)) = TIME
                    LET SIGNL(NOTE) = 0
             15     FILE NOTE IN CQUE
                    GO TO 11
             20     LET SASG(NOTE) = 1
                    LET RSRC(NOTE) = IRES
                    LET II = 0
                    DO TO 30, FOR EACH I IN SRHS(ESAC(NOTE))
                    IF SFLAG(I) EQ 1, LET II = 1
             30     LOOP
                    IF (II)EQ(0), LET SFLAG(NOTE) = 1
                    IF (SFLAG(NOTE))EQ(1), GO TO 40
                    CALL SSS(NOTE,IRES,2)
                    GO TO 11
             40     CALL SSS(NOTE,IRES,1)
                    GO TO 11
            999     CAUSE SRISE AT TIME + 1.0
                    RETURN
                    END
```

```
@ ELT XSET,1,710519, 63781

                ENDOGENOUS EVENT XSET
        C
        C       XSET OCCURS AT SUNSET EACH DAY. IT PLACES QUEUED 'LONG' SEARCH
        C       CASES IN THE SUNRISE LIST--FIFO SET LIST
        C
                IF CQUE IS EMPTY, GO TO 20
                DO TO 10, FOR EACH I IN CQUE
                IF (FLG(I))NE(3), GO TO  10
                REMOVE I FROM CQUE
                LET IK = SIGNL(I)
                LET SIGNL(I) = 2
                LET JIND = 2
                LET KIND = 2
                DO, FOR EACH J IN SRHS(ESAC(I))
                IF (SIGNL(J))EQ(0), LET JIND = 1
                IF (SIGNL(J))EQ(1), LET KIND = 1
                LOOP
                IF (IK)EQ(1), GO TO 50
                IF (JIND)EQ(1), GO TO 5
                IF (KIND)EQ(1), GO TO 5
                LET TQUE(ESAC(I)) = TQUE(ESAC(I)) + (TIME-TINQ(ESAC(I)))
                GO TO 5
           50   IF (KIND)EQ(1), GO TO 5
                IF (JIND)EQ(2), GO TO 60
                LET TINT(ESAC(I)) = TINT(ESAC(I)) + (TIME-STINQ(ESAC(I)))
                GO TO 5
           60   LET TQUE(ESAC(I)) = TQUE(ESAC(I)) + (TIME-TINQ(ESAC(I)))
                LET TINT(ESAC(I)) = TINT(ESAC(I)) + (TIME-STINQ(ESAC(I)))
            5   FILE I IN LIST
           10   LOOP
           20   CAUSE XSET AT TIME + 1.0
                RETURN
                END
```

```
000001          SUBROUTINE SERVE (CASE,INOTF,IRS)                                85000
000002          LET COSTC(CASE) = COSTC(CASE) + COST(IRS)                        85070
000003          LET CNRES(CASE) = CNRES(CASE) + 1                                85080
000004          LET NEEDS(STN(IRS)) = NEEDS(STN(IRS)) + 1                        85082
000005          LET NCASE(IRS) = NCASE(IRS) + 1
000006      158 LET IB(IRS) = 1                                                  85560
000007          IF INOTF NE 0, GO TO 50
000008          LET PRIOR(IRS) = PRI(CASE)                                       85600
000009          GO TO 60
000010       50 LET PRIOR(IRS) = PRI(INOTF)
000011          LET KRES(INOTF) = IRS
000012       60 IF EIAT(IRS) NE 0., GO TO (100,102,104,106,108,110,112,114,140,
000013         . 140,116,118,120),MFLG(IFLT(IRS))
000014          LET EIAT(IRS) = 99999.                                           85570
000015          CREATE FLT CALLED IFLT(IRS)                                      85130
000016          LET TFLT(IFLT(IRS)) = 0.0
000017          LET TOW(IFLT(IRS)) = 0                                           85160
000018           LET HCREW(IFLT(IRS)) = 0
000019          LET ACASE(IFLT(IRS)) = CASE
000020          LET FITON(IFLT(IRS)) = INOTF
000021           LET ROS(IFLT(IRS)) = TIME + TVEC(IRS) + DLAY(TYPE(IRS))
000022          IF INOTF NE 0, LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(INOTF)
000023          IF INOTF EQ 0, LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(CASE)
000024          IF IRS EQ -IS2(CASE), LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) +
000025         .TSM(CASE)/SOA3(TYPE(IRS))
000026          LET DEP(IFLT(IRS)) = 0.
000027          CREATE DELAY CALLED IDEV(IFLT(IRS))
000028          STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000029          STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000030          CAUSE DELAY CALLED IDEV(IFLT(IRS)) AT TIME + DLAY(TYPE(IRS))
000031          LET MFLG(IFLT(IRS)) = 11
000032          RETURN
000033      100 CANCEL ARVSN CALLED IDEV(IFLT(IRS))                              85180
000034          LET XR(IRS) = XI(IFLT(IRS))
000035          LET YR(IRS) = YI(IFLT(IRS))
000036          GO TO 145                                                        85190
000037      102 CANCEL ONSCN CALLED IDEV(IFLT(IRS))                              85200
000038          DESTROY ONSCN CALLED IDEV(IFLT(IRS))                             85210
000039          GO TO 140                                                        85220
000040      104 CANCEL ARSCH CALLED IDEV(IFLT(IRS))                              85230
000041          DESTROY ARSCH CALLED IDEV(IFLT(IRS))                             85240
000042          LET XR(IRS) = XI(IFLT(IRS))
000043          LET YR(IRS) = YI(IFLT(IRS))
000044          GO TO 140                                                        85250
000045      106 CANCEL COMPL CALLED IDEV(IFLT(IRS))                              85260
000046          DESTROY COMPL CALLED IDEV(IFLT(IRS))                             85270
000047          GO TO 140                                                        85280
000048      108 CANCEL SSET CALLED IDEV(IFLT(IRS))                               85290
000049          DESTROY SSET CALLED IDEV(IFLT(IRS))                              85300
000050          GO TO 140                                                        85310
000051      110 CANCEL FUEL CALLED IDEV(IFLT(IRS))                               85320
000052          DESTROY FUEL CALLED IDEV(IFLT(IRS))                              85330
000053          GO TO 140                                                        85340
000054      112 CANCEL HOME CALLED IDEV(IFLT(IRS))                               85350
```

```
000055          DESTROY HOME CALLED IDEV(IFLT(IRS))                                85360
000056          LET XR(IRS) = XI(IFLT(IRS))
000057          LET YR(IRS) = YI(IFLT(IRS))
000058          GO TO 140                                                          85370
000059      114 CANCEL HOMEF CALLED IDEV(IFLT(IRS))                                85380
000060          DESTROY HOMEF CALLED IDEV(IFLT(IRS))                               85390
000061          GO TO 140
000062      116 CANCEL DELAY CALLED IDEV(IFLT(IRS))
000063          DESTROY DELAY CALLED IDEV(IFLT(IRS))
000064          GO TO 140
000065      118 CANCEL READY CALLED IDEV(IFLT(IRS))
000066          DESTROY READY CALLED IDEV(IFLT(IRS))
000067          GO TO 140
000068      120 CANCEL CHEKN CALLED IDEV(IFLT(IRS))
000069          DESTROY CHEKN CALLED IDEV(IFLT(IRS))
000070      140 IF IWAIT(ACASE(IFLT(IRS))) EQ IRS, LET IWAIT(ACASE(IFLT(IRS))) = 0   85490
000071          CREATE ARVSN CALLED IDEV(IFLT(IRS))                                85500
000072      145 LET ACASE(IFLT(IRS)) = CASE                                        85510
000073          LET DEP(IFLT(IRS)) = TIME                                          85520
000074      155 LET XDEST(IFLT(IRS)) = XC(CASE)                                    85540
000075          LET YDEST(IFLT(IRS)) = YC(CASE)                                    85550
000076          LET TRLS = TIME + TVEC(IRS)
000077          LET ROS(IFLT(IRS)) = TRLS
000078          IF INOTF NE 0, GO TO 160
000079          LET TRLS = TIME + TVEC(IRS) + OST(CASE)
000080          LET RLS(IFLT(IRS)) = TRLS
000081          IF IRS EQ -IS2(CASE), LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) +        85620
000082         *TSM(CASE)/SOA3(TYPE(IRS))                                          85630
000083          LET FITON(IFLT(IRS)) = 0                                           85640
000084          GO TO 165                                                          85650
000085      160 LET RLS(IFLT(IRS)) = ROS(IFLT(IRS)) + OST(INOTF)
000086          LET FITON(IFLT(IRS)) = INOTF                                       85680
000087      165 STORE CASE IN CASNO(IDEV(IFLT(IRS)))                               85690
000088          STORE IRS IN RESNO(IDEV(IFLT(IRS)))                                85700
000089          CAUSE ARVSN CALLED IDEV(IFLT(IRS)) AT TIME + TVEC(IRS)
000090          LET MFLG(IFLT(IRS)) = 1                                            85740
000091          RETURN                                                             85750
000092          END                                                                85760
```

@ ELT DELAY,1,710520, 41408

```
000001                  ENDOGENOUS EVENT DELAY
000002                  STORE CASNO(DELAY) IN CASE
000003                  STORE RESNO(DELAY) IN IRS
000004                  DESTROY DELAY CALLED IDEV(IFLT(IRS))
000005                  LET XDEST(IFLT(IRS)) = XC(CASE)
000006                  LET YDEST(IFLT(IRS)) = YC(CASE)
000007                  LET DEP(IFLT(IRS)) = TIME
000008                  LET TFLT(IFLT(IRS)) = TIME
000009                  LET TEMP = ROS(IFLT(IRS))
000010                  IF TEMP LT TIME, LET TEMP = TIME
000011                  CREATE ARVSN CALLED IDEV(IFLT(IRS))
000012                  STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000013                  STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000014                  CAUSE ARVSN CALLED IDEV(IFLT(IRS)) AT TEMP
000015                  LET MFLG(IFLT(IRS)) = 1
000016                  RETURN
000017                  END OF DELAY
```

```
000001                ENDOGENOUS EVENT ARVS                                    90000
000002                STORE CASNO(ARVSN) IN CASE                               90010
000003                 STORE RESNO(ARVSN) IN IRS                               90020
000004                LET INOTF = FITON(IFLT(IRS))                             90030
000005                LET KFLAG = 0
000006                DESTROY ARVSN CALLED IDEV(IFLT(IRS))                     90100
000007            999 IF INOTF EQ 0, GO TO 12
000008                IF RESA(INOTF) EQ 0, LET RESA(INOTF) = IRS
000009             12 IF ITOL(CASE) NE 2, GO TO 50
000010                LET RESA(CASE) = IRS
000011                LET TWAIT(CASE) = TIME - OCCUR(CASE)                     90200
000012                LET NCAS(STN(IRS)) = NCAS(STN(IRS)) + 1                  90202
000013                LET AVGTW(STN(IRS)) = AVGTW(STN(IRS)) + TWAIT(CASE)      90203
000014                CALL STATS(CASE,IRS)
000015                IF INOTF EQ 0, GO TO 35
000016                IF NSET(CASE) IS EMPTY, GO TO 35
000017                DO TO 30, FOR EACH NOTIF IN NSET(CASE)
000018                IF SIGNL(NOTIF) NE 2, GO TO 40
000019             30 LOOP
000020             35 LET TQUEI(CASE) = TQUE(CASE)
000021                GO TO 42
000022             40 LET TQUEI(CASE) = TQUE(CASE) + TIME-TINQ(CASE)
000023             42 IF TWAIT(CASE) LE TOL(PRI(CASE)), GO TO 45
000024                LET FAIL3(STN(IRS)) = FAIL3(STN(IRS)) + 1
000025                LET ITOL(CASE) = 0                                       90230
000026                GO TO 50                                                 90240
000027             45 LET ITOL(CASE) = 1                                       90250
000028             50 LET DEP(IFLT(IRS)) = 0.0
000029                LET XR(IRS) = XDEST(IFLT(IRS))                           90110
000030                LET YR(IRS) = YDEST(IFLT(IRS))                           90120
000031                LET XDEST(IFLT(IRS)) = 0.                                90130
000032                LET YDEST(IFLT(IRS)) = 0.                                90140
000033                IF IWAIT(CASE) EQ 0, GO TO 300
000034                IF INOTF EQ 0, GO TO 301
000035                LET IR = IWAIT(CASE)                                     90440
000036                IF NUMBR(INOTF) NE 0, GO TO 299                          90442
000037                IF RLS(IFLT(IRS)) LS RLS(IFLT(IR)), GO TO 301
000038            299 IF IR EQ IRS, GO TO 301
000039                LET IB(IR) = 1
000040                IF MFLG(IFLT(IR)) EQ 2, GO TO 300                        90470
000041                IF MFLG(IFLT(IR)) NE 13, GO TO 298
000042                CANCEL CHEKN CALLED IDEV(IFLT(IR))
000043                DESTROY CHEKN CALLED IDEV(IFLT(IR))
000044                LET MFLG(IFLT(IR)) = 9
000045            298 CALL RETN(CASE,IR)
000046            300 LET IWAIT(CASE) = IRS                                    90490
000047                LET IB(IRS) = 2                                          90497
000048            301 LET D = 0.                                               90492
000049            302 IF MMM(CASE) + NNN(CASE) GR 1, GO TO 310
000050        C       SINGLE RESOURCE CASE                                     90510
000051                IF MMM(CASE) NE 0, GO TO 330                             90520
000052        C       SINGLE RESOURCE NON-TOM CASE                             90550
000053                GO TO 315                                                90560
000054        C       MULTI-RESOURCE CASE                                      90570
```

```
000055          310 IF NUMBR(INOTF) NE 0, GO TO 370                                90600
000056        C     MULTI-RESOURCE, NO.-TOW CASE                                   90610
000057          315 LET JFLAG = 1
000058              GO TO 600
000059          316 IF KFLAG EQ 1, GO TO 317
000060              CREATE ONSCN CALLED IDEV(IFLT(IRS))
000061              STORE CASE IN CASNO(IDEV(IFLT(IRS)))                           90630
000062              STORE IRS IN RESNO(IDEV(IFLT(IRS)))                            90640
000063              CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TIME                     90650
000064          317 LET MFLG(IFLT(IRS)) = 2
000065              IF INOTF GR 0, LET COMP(INOTF) = 2                             90670
000066              RETURN                                                        110370
000067        C     SINGLE RESOURCE CASE  TOE OR ESCORT                            90730
000068          330 CALL DTD(CASE,*XDEST(IFLT(IRS)),*YDEST(IFLT(IRS)),*D)          90740
000069          350 IF NEED(CASE) NE 17, GO TO 360                                 90750
000070        C     SINGLE RESOURCE - AIR ESCORT                                   90760
000071              LET OST(CASE) = D/SOA1(TYPE(IRS))
000072              LET TOWSP(IFLT(IRS)) = SOA1(TYPE(IRS))
000073              LET DEP(IFLT(IRS)) = TIME                                      90742
000074              GO TO 367
000075        C     SINGLE RESOURCE CASE TOW - NOT AIR ESCORT                      90800
000076          360 LET DEP(IFLT(IRS)) = TIME + THOOK
000077              IF L(CASE) LE 26, GO TO 365
000078              LET OST(CASE) = D/TSP2+THOOK
000079              LET TOWSP(IFLT(IRS)) = TSP2
000080              GO TO 367                                                      90830
000081          365 LET OST(CASE) = D/TSP1+THOOK
000082              LET TOWSP(IFLT(IRS)) = TSP1
000083          367 LET JFLAG = 2
000084              GO TO 600
000085          368 LET TRLS = TIME + OST(CASE)
000086              LET RLS(IFLT(IRS)) = TRLS
000087              GO TO 500                                                      90860
000088        C     MULTI-RESOURCE TOW OR ESCORT                                   90870
000089          370 LET D = SQRT((XC(CASE)-XHAND(INOTF))**2+(YC(CASE)-YHAND(INOTF))  90880
000090             **2)                                                            90890
000091              LET XDEST(IFLT(IRS)) = XHAND(INOTF)                            90897
000092              LET YDEST(IFLT(IRS)) = YHAND(INOTF)                            90898
000093              IF NEED(INOTF) NE 17, GO TO 390                                90900
000094              LET OST(INOTF) = D/SOA1(TYPE(IRS))
000095              LET TOWSP(IFLT(IRS)) = SOA1(TYPE(IRS))
000096              LET DEP(IFLT(IRS)) = TIME                                      90899
000097              GO TO 395                                                      90920
000098          390 LET DEP(IFLT(IRS)) = TIME + THOOK
000099              IF L(CASE) LE 26, GO TO 392
000100              LET OST(INOTF) = D/TSP2 + THOOK                                90930
000101              LET TOWSP(IFLT(IRS)) = TSP2
000102              GO TO 395                                                      90940
000103          392 LET OST(INOTF) = D/TSP1 + THOOK                                90960
000104              LET TOWSP(IFLT(IRS)) = TSP1
000105          395 LET JFLAG = 3
000106              GO TO 600
000107          396 LET TRLS = TIME + OST(INOTF)
000108              LET RLS(IFLT(IRS)) = TRLS
000109          500 IF KFLAG EQ 1, GO TO 515
000110              CREATE ONSCN CALLED IDEV(IFLT(IRS))
000111              STORE CASE IN CASNO(IDEV(IFLT(IRS)))                           90990
000112              STORE IRS IN RESNO(IDEV(IFLT(IRS)))                            91000
```

```
000113              CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TRLS
000114          515 LET TOW(IFLT(IRS)) = 1
000115              LET MFLG(IFLT(IRS)) = 2                                  91060
000116              IF INOTF GR 0, LET COMP(INOTF) = 2                       91090
000117              RETURN                                                   91100
000118          600 IF S2S(CASE) EQ 0, GO TO 650
000119              IF LOC(CASE) EQ 1, GO TO 650
000120        C                                                              90280
000121        C     A SHORT SEARCH IS NEEDED TO LOCATE THE CASE.             90290
000122        C                                                              90300
000123              CALL SRCHF(CASE,INOTF,IRC)
000124              LET PRI(CASE) = S2PRI                                    90310
000125              IF INOTF EQ 0, GO TO 70                                  90320
000126              DO TO 60, FOR EACH N IN NSET(CASE)                       90330
000127              IF KRES(N) NE 0, LET PRIOR(KRES(N)) = PRI(CASE)          90340
000128              LET PRI(N) = PRI(CASE)                                   90350
000129           60 LOOP                                                     90360
000130              GO TO 100
000131           70 LET PRIOR(IRS) = PRI(CASE)                               90380
000132          100 LET LOC(CASE) = 1                                        90420
000133              IF IRS NE IS2(CASE), GO TO 650
000134              LET KFLAG = 1
000135              CREATE ONSCN CALLED IDEV(IFLT(IRS))
000136              STORE CASE IN CASNO(IDEV(IFLT(IRS)))
000137              STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000138              LET TRLS = TIME+TSM(CASE)/SOA3(TYPE(IRS))
000139              CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TRLS
000140          650 GO TO (316,368,396),JFLAG
000141              END                                                      91110
```

```
000001                SUBROUTINE STATS(CASE,IRS)
000002          C
000003          C     SUBROUTINE STATS COLLECTS CASE STATISTICS, STATION STATISTICS
000004          C     AND GROUP STATISTICS WHEN THE FIRST RESOURCE ARRIVES AT THE
000005          C     SCENE OF A CASE.
000006          C
000007                LET DMERT(CASE) = 0.0
000008                LET X = TWAIT(CASE) - TOL(PRI(CASE))
000009                IF (X)LE(0.0), GO TO 10
000010                LET TWTOL(STN(IRS)) = TWTOL(STN(IRS)) + X
000011                LET MNTMT = MNTMT + X
000012                LET TMTAV(GRP(STN(IRS))) = TMTAV(GRP(STN(IRS))) + X
000013                IF PRI(CASE) GR 2, GO TO 40
000014                IF OFSHR(CASE) LE 20.0, LET BQ = 2.0
000015                IF OFSHR(CASE) GR 20.0, LET BQ = 1.0
000016                GO TO 41
000017             40 IF OFSHR(CASE) LE 20.0, LET BQ = 3.0
000018                IF OFSHR(CASE) GR 20.0, LET BQ = 2.0
000019             41 LET DMERT(CASE) = BQ * X
000020                LET DMRT(STN(IRS)) = DMRT(STN(IRS)) + DMERT(CASE)
000021                LET MEAND = MEAND + DMERT(CASE)
000022                LET AVDRT(GRP(STN(IRS))) = AVDRT(GRP(STN(IRS))) + DMERT(CASE)
000023             10 LET TWAVG(GRP(STN(IRS))) = TWAVG(GRP(STN(IRS))) + TWAIT(CASE)
000024          C
000025          C     CALCULATE TVEC(IRS) TO COLLECT STATISTICS ON IT.
000026          C
000027                LET D = XC(CASE) - XR(IRS)
000028                LET Y = YC(CASE) - YR(IRS)
000029                LET D = SQRT(D*D+Y*Y)
000030                IF (SMELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 20
000031                LET TVEC(IRS) = D/SOA1(TYPE(IRS))
000032                GO TO 30
000033             20 LET TVEC(IRS) = D/SOA2(TYPE(IRS))
000034             30 LET VCTR(STN(IRS)) = VCTR(STN(IRS)) + TVEC(IRS)
000035                LET MEANV = MEANV + TVEC(IRS)
000036                LET TVAVG(GRP(STN(IRS))) = TVAVG(GRP(STN(IRS))) + TVEC(IRS)
000037          C
000038          C     DETERMINE IF IT IS A WEEKDAY OR WEEKEND.
000039          C
000040                LET I = 3
000041                IF (PSHFT)GR(NWD), LET I = 4
000042                LET MEEN(I) = MEEN(I) + TWAIT(CASE)
000043                LET MEEN(I+2) = MEEN(I+2) + TVEC(IRS)
000044                LET MEEN(I+4) = MEEN(I+4) + X
000045                IF X GR 0.0, LET MEEN(I+6) = MEEN(I+6) + X
000046                LET CNTR(I) = CNTR(I) + 1
000047                LET CNTR(I+2) = CNTR(I+2) + 1
000048                LET CNTR(I+4) = CNTR(I+4) + 1
000049                LET CNTR(I+6) = CNTR(I+6) + 1
000050                LET STDEV(I) = STDEV(I)+(TWAIT(CASE)*TWAIT(CASE))
000051                LET STDEV(I+2)=STDEV(I+2)+(TVEC(IRS)*TVEC(IRS))
000052                LET STDEV(I+4) = STDEV(I+4) + (X*X)
000053                IF X GR 0.0, LET STDEV(I+6) = STDEV(I+6) + (X*X)
000054                LET K = 0
```

```
000055          GO TO 100, FOR J=(1)(4)
000056          IF (J)EQ(1), LET Y = TAMIT(CASE) * 24.0
000057          IF (J)EQ(2), LET Y = TVFC(IRS) * 24.0
000058          IF (J)GE(3), LET Y = X*24.0
000059          IF (J)LE(2), GO TO 101
000060          IF (Y)LE(0.0), GO TO 100
000061     101  IF Y GR 0.5, GO TO 102
000062          LET CATG1(I+K) = CATG1(I+K) + 1
000063          GO TO 99
000064     102  IF Y GR 1.0, GO TO 103
000065          LET CATG2(I+K) = CATG2(I+K) + 1
000066          GO TO 99
000067     103  IF Y GR 2.0, GO TO 104
000068          LET CATG3(I+K) = CATG3(I+K) + 1
000069          GO TO 99
000070     104  IF Y GR 3.0, GO TO 105
000071          LET CATG4(I+K) = CATG4(I+K) + 1
000072          GO TO 99
000073     105  IF Y GR 4.0, GO TO 106
000074          LET CATG5(I+K) = CATG5(I+K) + 1
000075          GO TO 99
000076     106  IF Y GR 5.0, GO TO 107
000077          LET CATG6(I+K) = CATG6(I+K) + 1
000078          GO TO 99
000079     107  IF Y GR 10.0, GO TO 108
000080          LET CATG7(I+K) = CATG7(I+K) + 1
000081          GO TO 99
000082     108  LET CATG8(I+K) = CATG8(I+K) + 1
000083      99  LET K = K + 2
000084     100  LOOP
000085          RETURN
000086          END
```

```
@ ELT SRCHF,1,71n520, 414n9

000001                SUBROUTINE SRCHF(CASE,INOTF,IRS)
000002                IF S2S(CASE) EQ 1, GO TO 200                                  110050
000003             15 LET SMIL = TSM(CASE)                                          110060
000004                CALL SASS(CASE,SMIL,IDUN,*IND)                                110070
000005                IF IND NE 0, GO TO 200
000006             50 LET IRAP = RAP                                                110140
000007                IF RAP EQ 4, LET IRAP = 5                                     110150
000008                IF RAP EQ 5, LET IRAP = 6                                     110160
000009                CALL RESAP(CASE,IRAP,*JRS)
000010                IF JRS EQ 0, GO TO 200
000011            100 LET IS2(CASE) = -JRS                                          110230
000012                IF MMM(CASE) + NNN(CASE) LE 1, GO TO 150                      110240
000013                CREATE NOTIF                                                  110250
000014                FILE NOTIF IN NSET(CASE)                                      110260
000015                LET COMP(NOTIF) = 2
000016                LET KRES(NOTIF) = JRS                                         110280
000017                LET PRI(NOTIF) = PRI(CASE)                                    110290
000018                LET CAS(NOTIF) = CASE                                         110300
000019                LET XHAND(NOTIF) = 0.                                         110320
000020                LET YHAND(NOTIF) = 0.                                         110330
000021                LET NEED(NOTIF) = 18                                          110331
000022                LET RESA(NOTIF) = 0                                           110332
000023                LET NUMBR(NOTIF) = 0                                          110333
000024                LET FLG(NOTIF) = 2
000025                LET SIGNL(NOTIF) = 2
000026                LET OST(NOTIF) = TSM(CASE)/SOA3(TYPE(JRS))                    110340
000027                LET OST(INOTF)=OST(INOTF) + DVEC(JRS) + TSM(CASE)/SOA3(TYPE(JRS))  110350
000028                IF NUMBR(INOTF) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000029               * DVEC(JRS) + OST(NOTIF)
000030                LET RLS(IFLT(IRS)) = TIME + OST(INOTF)
000031                CALL SERVE (CASE,NOTIF,JRS)                                   110360
000032                IF RLS(IFLT(IRS)) LE RLS(IFLT(JRS)), LET RLS(IFLT(IRS)) =
000033               *RLS(IFLT(IRS)) + 2.0/1440.0
000034                LET PRIOR(JRS) = S2PRI
000035            999 RETURN
000036            150 CALL SERVE(CASE,0,JRS)                                        110380
000037                LET PRIOR(JRS) = S2PRI
000038                IF MMM(CASE) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000039               * DVEC(JRS) + TSM(CASE)/SOA3(TYPE(JRS))
000040                LET TRLS = TIME + OST(CASE) + DVEC(JRS)+TSM(CASE)/SOA3(TYPE(JRS))
000041                LET RLS(IFLT(IRS)) = TRLS
000042                LET OST(CASE) = OST(CASE) + TSM(CASE)/SOA3(TYPE(JRS)) + DVEC(JRS)
000043                RETURN                                                        110390
000044            C                                                                 110400
000045            C       SEARCH IS DONE BY THE FIRST RESOURCE TO ARRIVE ON SCENE   110410
000046            C                                                                 110420
000047            200 LET IS2(CASE) = IRS                                           110430
000048                IF MMM(CASE) + NNN(CASE) LE 1, GO TO 250                      110440
000049                LET OST(INOTF) = OST(INOTF) + TSM(CASE)/SOA3(TYPE(IRS))
000050                LET TRLS = TIME + OST(INOTF)
000051                LET RLS(IFLT(IRS)) = TRLS
000052                IF NUMBR(INOTF) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000053               * TSM(CASE)/SOA3(TYPE(IRS))
000054                RETURN                                                        110460
```

```
000055          250 LET OST(CASE) = OST(CASE) + TSM(CASE)/SOA3(TYPE(IRS))          110470
000056              LET TRLS = TIME + OST(CASE)
000057              LET RLS(IFLT(IRS)) = TRLS
000058              IF MMM(CASE) NE 0, LET DEP(IFLT(IRS)) = DEP(IFLT(IRS)) +
000059             * TSM(CASE)/SOA3(TYPE(IRS))
000060              RETURN                                                          110480
000061              END                                                             110490
```

```
000001          ENDOGENOUS EVENT ONSCN                                          95000
000002          STORE CASNO(ONSCN) IN CASE                                      95001
000003          STORE RESNO(ONSCN) IN IRS                                       95002
000004          LET INOTF = FITON(IFLT(IRS))                                    95003
000005          IF IRS NE IS2(CASE), GO TO 9
000006          LET S2S(CASE) = -S2S(CASE)
000007          LET PRI(CASE) = FPRI(CASE)
000008          IF MMM(CASE) + NNN(CASE) GR 1, GO TO 3
000009          LET PRIOR(IRS) = FPRI(CASE)
000010          GO TO 5
000011        3 DO TO 4, FOR EACH N IN NSET(CASE)
000012          LET PRI(N) = PRI(CASE)
000013          IF KRES(N) NE 0, LET PRIOR(KRES(N)) = PRI(CASE)
000014        4 LOOP
000015        5 LET TRLS = RLS(IFLT(IRS)) - TIME
000016          IF TRLS LS 0., LET TRLS = 0.
000017          IF INOTF GR 0, LET OST(INOTF) = TRLS
000018          IF INOTF EQ 0, LET OST(CASE) = TRLS
000019          LET IS2(CASE) = 0
000020          IF INOTF EQ 0, GO TO 7
000021          IF NUMBR(INOTF) NE 0, GO TO 50
000022          GO TO 9
000023        7 IF MMM(CASE) NE 0, GO TO 50
000024        9 LET PRD = OST(CASE)/KKK
000025          IF INOTF GR 0, LET PRD = OST(INOTF)/KKK                         95030
000026          IF IRS EQ -IS2(CASE), LET PRD = TSM(CASE)/(SOA3(TYPE(IRS))*KKK) 95040
000027          IF TIME GE RLS(IFLT(IRS)), GO TO 60                             95050
000028          IF S2S(CASE) GR 0, GO TO 40
000029          LET TEMP = RANDM                                                95060
000030          IF TEMP GR PRUP, GO TO 10                                       95070
000031          IF PRI(CASE) EQ 5, GO TO 40                                     95080
000032          LET PRI(CASE) = PRI(CASE) + 1                                   95090
000033          GO TO 15
000034       10 IF TEMP GR PRDN+PRUP, GO TO 40
000035          IF PRI(CASE) EQ 1, GO TO 40                                     95120
000036          LET PRI(CASE) = PRI(CASE) - 1                                   95130
000037       15 IF INOTF GR 0, GO TO 20
000038          LET PRIOR(IRS) = PRI(CASE)                                      95150
000039          GO TO 40                                                        95160
000040       20 DO TO 30, FOR EACH N IN NSET(CASE)                              95170
000041          LET PRI(N) = PRI(CASE)                                          95180
000042          IF KRES(N) GR 0, LET PRIOR(KRES(N)) = PRI(CASE)
000043       30 LOOP                                                            95200
000044       40 IF TIME +PRD GE RLS(IFLT(IRS)), GO TO 50                        95210
000045          CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT TIME + PRD                95280
000046          RETURN                                                          95290
000047       50 CAUSE ONSCN CALLED IDEV(IFLT(IRS)) AT RLS(IFLT(IRS))            95300
000048          RETURN                                                          95370
000049       60 DESTROY ONSCN CALLED IDEV(IFLT(IRS))                            95380
000050          LET TOW(IFLT(IRS)) = 0                                          95410
000051          LET DEP(IFLT(IRS)) = 0.                                         95414
000052          LET MFLG(IFLT(IRS)) = 9                                         95420
000053          IF IRS EQ -IS2(CASE), GO TO 100
000054          LET COUNT(CASE) = COUNT(CASE) + 1                               95430
```

```
000055          IF COUNT(CASE) EQ MMM(CASE) + NNN(CASE), LET IB(IRS) = 1          95440
000056          IF MMM(CASE) EQ 0, GO TO 100                                      95450
000057          LET N1 = COUNT(CASE)-NNN(CASE) + 1                                95460
000058          IF N1 LE 0, GO TO 100                                             95470
000059          IF N1 EQ 1, GO TO 61
000060          LET XC(CASE) = XDEST(IFLT(IRS))
000061          LET YC(CASE) = YDEST(IFLT(IRS))
000062          LET XR(IRS) = XDEST(IFLT(IRS))
000063          LET YR(IRS) = YDEST(IFLT(IRS))
000064          LET XDEST(IFLT(IRS)) = 0.                                         95412
000065          LET YDEST(IFLT(IRS)) = 0.                                         95413
000066       61 IF N1 GR MMM(CASE), GO TO 100
000067          DO TO 70, FOR EACH NOTIF IN NSET(CASE)                            95490
000068          IF NUMBR(NOTIF) NE N1, GO TO 70                                   95500
000069          IF N1 NE 1, GO TO 65                                              95510
000070          CALL CRES(CASE,NEED(NOTIF),*KNT,*II)                              95520
000071          IF II EQ 0, GO TO 68                                              95530
000072          IF KNT EQ 0, GO TO 68                                             95540
000073          IF RQUE IS EMPTY, GO TO 68                                        95550
000074          DO TO 62, FOR EACH I IN RQUE                                      95560
000075          IF I NE IRS, GO TO 62                                             95570
000076          LET TVEC(IRS) = 0.                                                95580
000077          LET COST(IRS) = 0.                                                95590
000078          LET IB(IRS) = 1                                                   95600
000079          LET KRES(INOTF) = 0
000080          CALL SERVE(CASE,NOTIF,IRS)                                        95610
000081          RETURN                                                            95620
000082       62 LOOP                                                              95630
000083          GO TO 68                                                          95640
000084       65 LET OFSHR(CASE) = .25                                             95650
000085       68 CAUSE NOTIF AT TIME                                               95660
000086          GO TO 100                                                         95670
000087       70 LOOP                                                              95680
000088      100 IF IB(IRS) NE 2, GO TO 148
000089          IF CQUE IS EMPTY, GO TO 900
000090          DO TO 130, FOR EACH ICNN IN CQUE
000091          IF FLG(ICNN) NE 2, GO TO 130
000092          IF CAS(ICNN) NE CASE, GO TO 130
000093          CALL CRES(CASE,NEED(ICNN),*KONTR,*II)
000094          IF II EQ 0, GO TO 120
000095          IF KONTR EQ 0, GO TO 120
000096          DO, FOR EACH IRES OF RQUE
000097          IF IRES EQ IRS, GO TO 105
000098          LOOP
000099          GO TO 130
000100      105 LET TVEC(IRS) = 0.
000101          LET KRES(FITON(IFLT(IRS))) = 0
000102          CALL SVQUE(ICNN,IRS)
000103          RETURN
000104      120 LET ITOL(CASE) = 7
000105          CALL NRECK(ICNN)
000106      130 LOOP
000107      900 CREATE CHEKN CALLED IDEV(IFLT(IRS))
000108          STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000109          CAUSE CHEKN CALLED IDEV(IFLT(IRS)) AT TIME + TCHEK
000110          LET MFLG(IFLT(IRS)) = 13
000111          RETURN
000112      148 CALL RETN(CASE,IRS)
```

```
000113          150 RETURN                                                  95710
000114              END                                                     95720
```

```
000001          SUBROUTINE COVER (CASE,IRES)                                180000
000002          IF MMM(CASE) + NNN(CASE) EQ 1, GO TO 900                    180010
000003          LET IW = 0                                                  180020
000004          LET XRLS = 0.                                               180030
000005          DO TO 100, FOR EACH NOTI, IN NSET(CASE)                     180040
000006          LET IR = KRES(NOTIF)                                        180050
000007          IF IR EQ 0, GO TO 100                                       180060
000008          IF MFLG(IFLT(IR)) NE 2, GO TO 100                           180070
000009          IF IR EQ IRES, GO TO 100                                    180080
000010          IF RLS(IFLT(IR)) LE XRLS, GO TO 100                         180090
000011          LET XRLS = RLS(IFLT(IR))                                    180100
000012          LET IW = IR                                                 180110
000013      100 LOOP                                                        180120
000014          IF IW EQ 0, GO TO 900                                       180130
000015          LET IWAIT(CASE) = IW                                        180140
000016          LET IB(IW) = 2                                              180150
000017          RETURN                                                      180160
000018      900 LET IWAIT(CASE) = 0                                         180170
000019          RETURN                                                      180180
000020          END                                                         180190
```

@ ELT CHEKN,1,71/520, 41439

```
000001                 ENDOGENOUS EVENT CHEKN
000002                 STORE RESNO(CHEKN) IN IRS
000003                 LET MFLG(IFLT(IRS)) = 9
000004                 DESTROY CHEKN CALLED IDEV(IFLT(IRS))
000005                 LET ICK = 1
000006                 CALL EXQ(IRS,**ICK)
000007                 IF ICK EQ 1, GO TO 10
000008                 RETURN
000009              10 CREATE CHEKN CALLED IDEV(IFLT(IRS))
000010                 STORE IRS IN RESNO(IDEV(IFLT(IRS)))
000011                 CAUSE CHEKN CALLED IDEV(IFLT(IRS)) AT TIME + TCHEK
000012                 LET MFLG(IFLT(IRS)) = 13
000013                 RETURN
000014                 END
```

```
@ ELT RETN,1,710620, 41439

000001                SUBROUTINE RETN(CASE,IRS,                                100000
000002            999 LET INOTF = FITON(IFLT(IRS))
000003              1 IF IB(IRS) EQ 2, RETURN
000004                LET IB(IRS) = 0                                          100070
000005                IF INOTF EQ 0, GO TO 20
000006                LET KRES(INOTF) = 0                                      100102
000007                LET COMP(INOTF) = 3                                      100110
000008             20 LET IVAR = 0
000009                CALL EXQ(IRS,**IVAR)
000010                IF COUNT(CASE) LT MMM(CASE)+NNN(CASE), GO TO 21          100180
000011                CALL TERM(CASE)                                          100190
000012                RETURN                                                   100200
000013             21 IF IRS NE -IS2(CASE), RETURN
000014                LET S2S(CASE) = -S2S(CASE)
000015                LET PRI(CASE) = FPRI(CASE)                               100220
000016                IF MMM(CASE) + NNN(CASE) GR 1, GO TO 30                  100230
000017             22 DO TO 25, FOR I=(1)(NRES)
000018                IF (IFLT(I)) EQ 0, GO TO 25                              100270
000019                IF ACASE(IFLT(I)) NE CASE, GO TO 25                      100280
000020                IF I EQ IRS, GO TO 25                                    100290
000021                LET PRIOR(I) = FPRI(CASE)                                100300
000022                LET IS2(CASE) = 0                                        100310
000023                RETURN                                                   100320
000024             25 LOOP                                                     100330
000025                RETURN                                                   100370
000026             30 DO TO 35, FOR EACH N IN NSET(CASE)                       100380
000027                LET PRI(N) = PRI(CASE)                                   100390
000028                IF KRES(N) NE 0, LET PRIOR(KRES(N)) = PRI(CASE)          100400
000029             35 LOOP                                                     100410
000030                LET IS2(CASE) = 0
000031             40 RETURN                                                   100420
000032                END                                                      100430
```

```
% ELT TERM,1,710520, 41439

000001                SUBROUTINE TERM(CASE)                                          105000
000002                DIMENSION IBUFF(15),FBUFF(10)
000003                LET NMBRQ(STATN(CASE)) = NMBRQ(STATN(CASE)) + NQUE(CASE)
000004                LET TSVC(CASE) = TIME-OCCUR(CASE)                              105010
000005                LET I = 11
000006                IF (PSHFT)GR(NWD), LET I = 12
000007                LET MEEN(I) = MEEN(I) + TSVC(CASE)
000008                LET STDEV(I) = STDEV(I) + (TSVC(CASE)*TSVC(CASE))
000009                LET CNTR(I) = CNTR(I) + 1
000010                LET Y = TSVC(CASE) * 24.0
000011                IF Y GR 0.5, GO TO 102
000012                LET CATG1(I) = CATG1(I) + 1
000013                GO TO 99
000014            102 IF Y GR 1.0, GO TO 103
000015                LET CATG2(I) = CATG2(I) + 1
000016                GO TO 99
000017            103 IF Y GR 2.0, GO TO 104
000018                LET CATG3(I) = CATG3(I) + 1
000019                GO TO 99
000020            104 IF Y GR 3.0, GO TO 105
000021                LET CATG4(I) = CATG4(I) + 1
000022                GO TO 99
000023            105 IF Y GR 4.0, GO TO 106
000024                LET CATG5(I) = CATG5(I) + 1
000025                GO TO 99
000026            106 IF Y GR 5.0, GO TO 107
000027                LET CATG6(I) = CATG6(I) + 1
000028                GO TO 99
000029            107 IF Y GR 10.0, GO TO 108
000030                LET CATG7(I) = CATG7(I) + 1
000031                GO TO 99
000032            108 LET CATG8(I) = CATG8(I) + 1
000033             99 IF PRTOT EQ 0, GO TO 10
000034                WRITE ON TAPE 6, CASE,NOCAS(CASE),TIME,OCCUR(CASE)
000035                FORMAT (S65,'CASE TERMINATED IS ',I6,' NOCAS=',I5,' AT TIME',
000036               .2M4.2.2)
000037                WRITE ON TAPE 6, NNN(CASE),MMM(CASE),S1S(CASE),S2S(CASE),
000038               .PRI(CASE),RESA(CASE),XC(CASE),YC(CASE),TSM(CASE),TQUE(CASE),
000039               .TINT(CASE)
000040                FORMAT (S65,6I4,2D4.2,D5.0,2M4.2.2)
000041             10 LET NBRCO = NBRCO + 1
000042                LET KOUNT = KOUNT - 1
000043                IF STAPE EQ 0, GO TO 29
000044                DO TO 11, FOR I=(1)(15)
000045                LET IBUFF(I) = 0
000046                IF I GR 10, GO TO 11
000047                LET FBUFF(I) = 0.
000048             11 LOOP
000049                WRITE ON TAPE STAPE, NBRCO,OPFAC(CASE),NOCAS(CASE),INLOC(CASE),
000050               .OCCUR(CASE),BOX(CASE),FPRI(CASE),MMM(CASE),NNN(CASE),GAMMA(CASE),
000051               .NEED(CASE),AIR(CASE),OFSHR(CASE),VIS(CASE),MIND(CASE),SWELL(CASE)
000052               .L(CASE),POB(CASE),S1S(CASE),S2S(CASE),TSM(CASE),OST(CASE),
000053               .DMERT(CASE)
000054                FORMAT (3I5,I3,D3.4,I5,I1,2I2,D3.2,I2,I5,D4.2,6I5,I2,D5.0,2D1.4)
```

```
000055          WRITE ON TAPE STAPE, WTYPE(CASE),VALUE(CASE),XCX(CASE),YCY(CASE),
000056         *XC(CASE),YC(CASE),STAT(CASE),CARFS(CASE),RESA(CASE),PRI(CASE),
000057         *REA(CASE),COSTC(CASE),ITOL(CASE),NOINT(CASE),WQUE(CASE),
000058         *TINT(CASE),IQUE(CASE),TQUE1(CASE),TSVC(CASE),TWAIT(CASE)
000059          FORMAT (I5,I10,F05.2,I5,I2,I3,2I1,D7.2,I1,2I2,5D3.4)
000060       29 LET I = 11
000061       30 IF SRHS(CASE) IS EMPTY, GO TO 39
000062          REMOVE FIRST N FROM SRHS(CASE)
000063          IF I LE 15, LET IBUFF(I) = RESA(N)
000064          LET I = I+1
000065          IF PRTOT EQ 0, GO TO 1
000066          WRITE ON TAPE 6, RESA(N)
000067          FORMAT (S65,I5)
000068        1 DESTROY NOTE CALLED N
000069          GO TO 30
000070       39 LET I=1
000071       40 IF NSET(CASE) IS EMPTY, GO TO 100
000072          REMOVE FIRST N FROM NSET(CASE)                                  105130
000073          IF PRTOT EQ 0, GO TO 2
000074          WRITE ON TAPE 6, RESA(N)
000075          FORMAT (S65,I5)
000076        2 IF I GR 10, GO TO 50
000077          LET IBUFF(I) = NEED(N)
000078          LET IBUFF(I+1) = RESA(N)
000079          LET FBUFF(I) = OST(N)
000080          LET FBUFF(I+1) = DELTA(N)
000081          LET I=I+2
000082       50 IF KRES(N) EQ 0, GO TO 52
000083          LET IB(KRES(N)) = 0
000084          LET IVV = 0
000085          CALL EXQ(KRES(N),**IVV)
000086       52 DESTROY NOTIF CALLED N
000087          GO TO 40
000088      100 DESTROY CASE                                                    105160
000089          IF STAPE EQ 0, RETURN
000090          WRITE ON TAPE STAPE,IBUFF(I),FBUFF(I),FBUFF(I+1),IBUFF(I+1), FOR
000091         * I = (1)(10)(2)
000092          FORMAT 5(I2,D1.4,D1.2,I3)
000093          WRITE ON TAPE STAPE, IBUFF(I), FOR I=(11)(15)
000094          FORMAT 5(I3)
000095          RETURN                                                          105170
000096          END                                                             105180
```

```
. ELT HOME,1,710520, 41409

000001          ENDOGENOUS EVENT HOME                                    115000
000002          STORE RESNO(HOME) IN IRS                                 115010
000003          LET EIAT(IRS) = 0.                                       115060
000004          DESTROY HOME CALLED IDEV(IFLT(IRS))
000005          LET BCREM(STN(IRS)) = BCREM(STN(IRS)) - 1
000006          IF TFLT(IFLT(IRS)) GR TLAST, GO TO 10
000007          LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST               115100
000008          GO TO 20                                                 115110
000009       10 LET TUTIL(IRS) = TUTIL(IRS) + TIME-TFLT(IFLT(IRS))       115120
000010       20 LET XR(IRS) = XS(STN(IRS))                               115130
000011          LET YR(IRS) = YS(STN(IRS))                               115140
000012        1 DESTROY FLT CALLED IFLT(IRS)
000013          LET IFLT(IRS) = 0
000014          RETURN                                                   115150
000015          END                                                      115160
```

```
000001                SUBROUTINE SAQ(IRS)
000002          C
000003          C     SUBROUTINE SAQ IS CALLED WHEN A CASE IS INTERRUPTED.  AN ATTEMPT
000004          C     IS MADE TO FIND AN IDLE RESOURCE TO SERVE IT; IF NONE IS FOUND,
000005          C     THE CASE (OR NEED) IS QUEUED.
000006          C
000007                LET JK = ACASE(IFLT(IRS))
000008                IF (TOW(IFLT(IRS)))NE(1), GO TO 1001
000009                LET TOW(IFLT(IRS)) = 0
000010                LET XC(JK) = XI(IFLT(IRS))
000011                LET YC(JK) = YI(IFLT(IRS))
000012                LET XR(IRS) = XI(IFLT(IRS))
000013                LET YR(IRS) = YI(IFLT(IRS))
000014           1001 LET NOINT(JK) = NOINT(JK) + 1
000015                LET TOTIN = TOTIN + 1
000016                LET NINTR(STN(IRS)) = NINTR(STN(IRS)) + 1
000017               IF IWAIT(JK) EQ IRS, CALL COVER(JK,IRS)                          75140
000018                IF (IB(IRS))NE(2), GO TO 1002
000019                IF (MFLG(IFLT(IRS)))EQ(2), GO TO 1002
000020                LET KRES(FITON(IFLT(IRS))) = 0
000021                RETURN
000022           1002 IF (SIS(JK))GR(0), GO TO 1030
000023                IF (MMM(JK)+NNN(JK))EQ(1), GO TO 1010
000024                GO TO 1020
000025          C
000026          C     A SINGLE RESOURCE CASE IS BEING INTERRUPTED; SEND IT BACK THROUG
000027          C     THE SYSTEM TO SEE IF AN IDLE RESOURCE CAN SERVE IT; IF NOT, QUEUE
000028          C     THE CASE.
000029          C
000030           1010 IF (NEED(JK))NE(17), GO TO 1014
000031                LET ITOL(JK) = 5
000032                FILE JK IN EXCS
000033                RETURN
000034           1014 CALL CRES(JK,NEED(JK),*,NT,*II)
000035                CALL VEC(JK)
000036                CALL OSET(JK,KNT)
000037                DO TO 1011, FOR EVERY IRES IN RQUE
000038                IF (IB(IRES))NE(0), GO TO 1011
000039                IF (EIAT(IRES))EQ(0.0), GO TO 1012
000040                IF (HCREW(IFLT(IRES)))EQ(0), GO TO 1013
000041                GO TO 1011
000042           1012 CALL ROCA(JK,IRES,*II)
000043                IF (II)EQ(1), GO TO 1013
000044           1011 LOOP
000045                GO TO 1015
000046           1013 CALL SERVE(JK,0,IRES)
000047                RETURN
000048           1015 IF (REA(JK))EQ(2), LET REA(JK) = 0
000049                LET I = PRI(JK) + IDELT
000050                IF (I)GR(5), GO TO 1016
000051                LET PRI(JK) = PRI(JK) + IDELT
000052           1016 LET TINQ(JK) = TIME
000053                LET SIGNL(JK) = 1
000054                LET NQUE(JK) = NQUE(JK) + 1
```

```
000055                FILE JK IN CQUE
000056                RETURN
000057          C
000058          C     A MULTI-RESOURCE NEED IS BEING INTERRUPTED; CALL SUBROUTINE QUEUE.
000059          C
000060           1020 CALL QUEUE(IRS,1)
000061                RETURN
000062          C
000063          C     A SEARCH NEED IS BEING INTERRUPTED; SEND IT BACK THROUGH THE
000064          C     SYSTEM TO SEE IF AN IDLE RESOURCE CAN SERVE IT; IF NOT, QUEUE
000065          C     THE CASE
000066          C
000067           1030 IF (MFLG(IFLT(IRS)))EQ(3), GO TO 1031
000068                IF (MFLG(IFLT(IRS)))EQ(12), GO TO 1031
000069                LET A = TIME - ROS(IFLT(IRS))
000070                LET A = A * SOA3(TYPE(IRS))
000071                LET SM(FITON(IFLT(IRS)))=SM(FITON(IFLT(IRS)))-A
000072           1031 LET KDT = FITON(IFLT(IRS))
000073                CALL SASS(JK,SM(KDT),KDT,*IDC)
000074                DO TO 1032, FOR EVERY IRES IN RQUE
000075                IF (IB(IRES))NE(0), GO TO 1032
000076                IF (EIAT(IRES))EQ(0.0), GO TO 1033
000077                IF (HCREW(IFLT(IRES)))EQ(0), GO TO 1036
000078                GO TO 1032
000079           1033 CALL ROCA(JK,IRES,*II)
000080                IF (II)EQ(1), GO TO 1034
000081           1032 LOOP
000082                GO TO 1035
000083           1036 CALL SSS(KDT,IRES,3)
000084                LET RSRC(KDT) = IRES
000085                RETURN
000086           1035 IF (REA(JK))EQ(2), LET REA(JK) = 0
000087                LET I = PRI(JK) + IDELT
000088                IF (I)GR(5), GO TO 1037
000089                LET PRI(JK) = PRI(JK) + IDELT
000090                DO, FOR EACH I IN SRHS(JK)
000091                LET PRI(I) = PRI(I) + IDELT
000092                IF (RSRC(I))NE(0), LET PRIOR(RSRC(I)) = PRIOR(RSRC(I)) + IDELT
000093                LOOP
000094           1037 LET NQUE(JK) = NQUE(JK) + 1
000095                LET I = 0
000096                LET J = 0
000097                DO, FOR EACH K IN SRHS(JK)
000098                IF (SIGNL(K))EQ(0), LET I = 1
000099                IF (SIGNL(K))EQ(1), LET J = 1
000100                LOOP
000101                IF (I+J)EQ(0), LET TINQ(JK) = TIME
000102                IF (J)EQ(0), LET STINQ(JK) = TIME
000103                LET SIGNL(FITON(IFLT(IRS))) = 1
000104                LET RSRC(FITON(IFLT(IRS))) = 0
000105                FILE FITON(IFLT(IRS)) IN CQUE
000106                RETURN
000107                END
```

```
000001                SUBROUTINE QUEUE(NCODE,M)                                125000
000002                LET NOTIF = NCODE                                        125010
000003                IF M EQ 1, LET NOTIF = FITON(IFLT(NCODE))                125020
000004                LET CASE = CAS(NOTIF)
000005                IF NEED(NOTIF) NE 17, GO TO 10                           125030
000006                LET ITOL(CASE) = 5                                       125040
000007                LET COMP(NOTIF) = 4
000008                CALL WRECK(NOTIF)                                        125050
000009                RETURN                                                   125060
000010             10 IF M EQ 0, GO TO 60                                      125070
000011          C                                                              125080
000012          C     INTERRUPT SITUATION, SEE IF SOME OTHER RESOURCE CAN SERVE THE CASE 125090
000013          C                                                              125100
000014                CALL CRES(CAS(NOTIF),NEED(NOTIF),*KONTR,*II)             125110
000015                IF II EQ 0, GO TO 20                                     125120
000016                IF KONTR EQ 0, GO TO 20                                  125130
000017                GO TO 30                                                 125140
000018             20 LET ITOL(CAS(NOTIF)) = 7                                 125150
000019                CALL WRECK(NOTIF)                                        125160
000020                RETURN                                                   125170
000021             30 CALL VEC(CAS(NOTIF))                                     125180
000022                CALL OSET(CAS(NOTIF),KONTR)                              125190
000023                IF RQUE IS EMPTY, GO TO 60                               125200
000024                DO TO 50, FOR EACH IRS IN RQUE                           125210
000025                IF IB(IRS) NE 0, GO TO 50                                125220
000026                IF EIAT(IRS) EQ 0., GO TO 40                             125230
000027                IF HCREW(IFLT(IRS)) NE 0, GO TO 50                       125240
000028                CALL SERVE(CAS(NOTIF),NOTIF,IRS)                         125250
000029                RETURN                                                   125260
000030             40  CALL ROCA(CAS(NOTIF),IRS,*IVAL)
000031                IF IVAL EQ 2, GO TO 50                                   125280
000032                CALL SERVE(CAS(NOTIF),NOTIF,IRS)                         125290
000033                RETURN                                                   125300
000034             50 LOOP                                                     125310
000035             60 LET N1 = 0                                               125320
000036                LET N2 = 0                                               125330
000037                DO TO 100, FOR EACH INOTE IN NSET(CAS(NOTIF))            125340
000038                IF COMP(INOTE) NE 1, GO TO 100                           125350
000039                LET N1 = N1 + 1                                          125360
000040                IF SIGNL(INOTE) EQ 0, GO TO 100                          125370
000041                LET N2 = N2 + 1                                          125380
000042            100 LOOP                                                     125390
000043          C                                                              125400
000044          C     N1 = NUMBER OF NOTIFS (FOR THIS CASE) ALREADY IN CQUE.   125410
000045          C     N2 = NUMBER OF NOTIFS IN CQUE BECAUSE OF INTERRUPTS.     125420
000046          C                                                              125430
000047                IF N1 EQ 0, LET TINQ(CAS(NOTIF)) = TIME                  125440
000048                IF N2 GR 0, GO TO 110                                    125450
000049                IF M EQ 0, GO TO 110                                     125460
000050                LET STINQ(CAS(NOTIF)) = TIME                             125470
000051            110 IF REA(CAS(NOTIF)) NE 2, GO TO 120                       125480
000052                LET REA(CAS(NOTIF)) = 1-M                                125490
000053            120 FILE NOTIF IN CQUE                                       125500
000054                LET KRES(NOTIF) = 0
```

```
000055          LET COMP(NOTIF) = 1                                          125510
000056          LET NQUE(CASE) = NQUE(CASE) + 1                              125520
000057          LET SIGNL(NOTIF) = M                                         125530
000058          IF M EQ 0, RETURN                                            125540
000059          LET PRI(CAS(NOTIF)) = PRI(CAS(NOTIF)) + IDELT                125550
000060          IF PRI(CAS(NOTIF)) GR 5, LET PRI(CAS(NOTIF)) = 5             125560
000061          DO TO 130, FOR EACH INOTF IN NSET(CASE)                      125570
000062          LET PRI(INOTF) = PRI(CAS(NOTIF))                             125580
000063          IF KRES(INOTF) NE 0, LET PRIOR(KRES(INOTF)) = PRI(CAS(NOTIF))125590
000064      130 LOOP                                                         125600
000065          RETURN                                                       125610
000066          END OF QUEUE                                                 125620
```

```
000001                SUBROUTINE EXQ(IRES,IVAR)
000002          C
000003          C     SUBROUTINE EXQ EXAMINES THE CASE QUEUE (CQUE) FOR A SINGLE RESOURCE
000004          C     CASE, A MULTI-RESOURCE NEED, OR A SEARCH NEED THAT CAN
000005          C     BE SERVED BY THE RESOURCE BECOMING IDLE.
000006          C
000007                IF (IVAR)EQ(0), GO TO 10
000008                IF CQUE IS NOT EMPTY, GO TO 20
000009                RETURN
000010          C
000011          C     COMPARE BCREW(STN(IRES)) WITH THE PRESENT SHIFT LEVEL.
000012          C
000013             10 LET K = SHIFT(STN(IRES),PSHFT) - BCREW(STN(IRES))
000014                IF (K)GE(0), GO TO 100
000015          C
000016          C     RESOURCE MUST BE VECTORED TO HOME STATION DUE TO CREW AVAILABILITY.
000017          C     CQUE IS NOT EXAMINED
000018          C
000019                LET HCREW(IFLT(IRES)) = 1
000020             15 LET IB(IRES) = 0
000021                LET DEP(IFLT(IRES)) = TIME
000022                LET XDEST(IFLT(IRES)) = XS(STN(IRES))
000023                LET YDEST(IFLT(IRES)) = YS(STN(IRES))
000024                LET X = XR(IRES) - XS(STN(IRES))
000025                LET Y = YR(IRES) - YS(STN(IRES))
000026                LET D = SQRT(X*X+Y*Y)
000027                IF (SWELL(ACASE(IFLT(IRES))))GR(SLIM(TYPE(IRES))), GO TO 16
000028                LET TVEC(IRES) = D/SOA1(TYPE(IRES))
000029                GO TO 17
000030             16 LET TVEC(IRES) = D/SOA2(TYPE(IRES))
000031             17 LET TARVL(IFLT(IRES)) = TIME + TVEC(IRES)
000032                CREATE HOME CALLED IDEV(IFLT(IRES))
000033                LET RESNO(IDEV(IFLT(IRES))) = IRES
000034                LET MFLG(IFLT(IRES)) = 7
000035                CAUSE HOME CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000036                RETURN
000037            100 IF CQUE IS EMPTY, GO TO 15
000038          C
000039          C     EXAMINE CASE QUEUE
000040          C
000041             20 DO TO 999, FOR EACH ICNN IN CQUE
000042                IF (FLG(ICNN))EQ(1), LET CASE = ICNN
000043                IF (FLG(ICNN))EQ(2), LET CASE = CAS(ICNN)
000044                IF (FLG(ICNN))EQ(3), LET CASE = ESAC(ICNN)
000045          C
000046          C     CHECK TO SEE IF THE STATION OF THE RESOURCE IS EQUAL TO ONE OF THE
000047          C     PRIMARY OR ADJACENT STATIONS OF THE CASE.
000048          C
000049            110 LET LSTA = STATN(CASE)
000050                IF (FLG(ICNN))EQ(2), GO TO 120
000051                IF (FLG(ICNN))EQ(3), GO TO 120
000052                GO TO (120,120,120,111,111,120), RAP
000053            111 IF (LSTA)EQ(STN(IRES)), GO TO 200
000054                DO TO 112, FOR I=(1)(NACS(LSTA))
```

```
000055                IF (STN(IRES))EQ(ACS(LSTA,I)), GO TO 200
000056          112   LOOP
000057                DO TO 113, FOR I=(1)(NCUT(LSTA))
000058                IF (STN(IRES))EQ(CUT(LSTA,I)), GO TO 200
000059          113   LOOP
000060                GO TO 999
000061          120   IF (LSTA)EQ(STN(IRES)), GO TO 200
000062                DO TO 121, FOR I=(1)(NADJS(LSTA))
000063                IF (STN(IRES))EQ(ADJS(LSTA,I)), GO TO 200
000064          121   LOOP
000065                DO TO 122, FOR I=(1)(NACS(LSTA))
000066                IF (STN(IRES))EQ(ACS(LSTA,I)), GO TO 200
000067          122   LOOP
000068                DO TO 123, FOR I=(1)(NCUT(LSTA))
000069                IF (STN(IRES))EQ(CUT(LSTA,I)), GO TO 200
000070          123   LOOP
000071                IF (ADJS(LSTA,1))EQ(0), GO TO 999
000072                DO TO 130, FOR I=(1)(NADJS(LSTA))
000073                DO TO 124, FOR KK=(1)(NACS(ADJS(LSTA,I)))
000074                IF (STN(IRES))EQ(ACS(ADJS(LSTA,I),KK)), GO TO 200
000075          124   LOOP
000076                DO TO 125, FOR KK=(1)(NCUT(ADJS(LSTA,I)))
000077                IF (STN(IRES))EQ(CUT(ADJS(LSTA,I),KK)), GO TO 200
000078          125   LOOP
000079          130   LOOP
000080                GO TO 999
000081          200   IF (FLG(ICNN))EQ(1), CALL CRES(ICNN,NEED(ICNN),*KONT,*II)
000082                IF (FLG(ICNN))EQ(2), CALL CRES(CAS(ICNN),NEED(ICNN),*KONT,*II)
000083                IF (FLG(ICNN))EQ(3), CALL CRES(ESAC(ICNN),18,*KONT,*II)
000084                DO TO 201,FOR EACH IR IN RQUE
000085                IF (IR)EQ(IRES), GO TO 210
000086          201   LOOP
000087                GO TO 999
000088          210   IF (SQTAG(TYPE(IRES)))EQ(0), GO TO 211
000089                IF (PRI(CASE))GE(TPRI), GO TO 211
000090                LET II = 1
000091                DO, FOR EACH IR IN RQUE
000092                IF (SQTAG(TYPE(IR)))EQ(0), LET II = 0
000093                LOOP
000094                IF (II)EQ(1), GO TO 211
000095                IF (FLG(ICNN))NE(3), GO TO 999
000096                DO TO 212, FOR EACH IR IN RQUE
000097                IF (SQTAG(TYPE(IR)))GR(0), GO TO 212
000098                LET X = XC(CASE) - XS(STN(IR))
000099                LET Y = YC(CASE) - YS(STN(IR))
000100                LET D = SQRT(X*X+Y*Y)
000101                IF (SMELL(CASE))GR(SLIM(TYPE(IR))), GO TO 700
000102                LET XY = D/SOA1(TYPE(IR))
000103                GO TO 701
000104          700   LET XY = D/SOA2(TYPE(IR))
000105          701   IF (END(TYPE(IR)))GR(2.0*XY), LET II = 1
000106          212   LOOP
000107                IF (II)EQ(1), GO TO 999
000108        C
000109        C       RESOURCE IS IN RQUE; CALCULATE TVEC(IRES)
000110        C
000111          211   LET X = XR(IRES) - XC(CASE)
000112                LET Y = YR(IRES) - YC(CASE)
```

```
000113                LET D = SQRT(X*X+Y*Y)
000114                IF (SMELL(CASE))GR(SLIM(TYPE(IRES))), GO TO 220
000115                LET TVEC(IRES) = D/SOA1(TYPE(IRES))
000116                GO TO 230
000117          220   LET TVEC(IRES) = D/SOA2(TYPE(IRES))
000118          230   IF (FLG(ICNN))EQ(2), GO TO 300
000119                IF (FLG(ICNN))EQ(3), GO TO 400
000120        C
000121        C       FLG(ICNN) = 1; I.E., A SINGLE RESOURCE IS BEING REMOVED FROM CQUE
000122        C       TO BE SERVED. ATTRIBUTES WILL BE UPDATED AND SUBROUTINE SERVE WILL
000123        C       BE CALLED.
000124        C
000125          250   IF (IVAR)EQ(0), GO TO 260
000126                LET IVAR = 0
000127                LET IWAIT(ACASE(IFLT(IRES))) = 0
000128                LET KRES(FITON(IFLT(IRES))) = 0
000129          260   LET TQUE(ICNN) = TQUE(ICNN) + (TIME-TINQ(ICNN))
000130                IF (SIGNL(ICNN))EQ(1),LET TINT(ICNN)=TINT(ICNN)+(TIME-TINQ(ICNN))
000131                LET COST(IRES) = COSTD(TYPE(IRES)) * TVEC(IRES)
000132                LET PRIOR(IRES) = PRI(ICNN)
000133                STORE ICNN IN ACASE(IFLT(IRES))
000134                REMOVE ICNN FROM CQUE
000135                CALL SERVE(ICNN,0,IRES)
000136                RETURN
000137        C
000138        C       FLG(ICNN) = 2; A MULTI-RESOURCE NEED IS BEING REMOVED FROM CQUE
000139        C       TO BE SERVED.  RES AND FLT ATTRIBUTES WILL BE UPDATED AND
000140        C       SUBROUTINE SVQUE WILL BE CALLED.
000141        C
000142          300   IF (IVAR)EQ(0), GO TO 301
000143                LET IVAR = 0
000144                LET IWAIT(ACASE(IFLT(IRES))) = 0
000145                LET KRES(FITON(IFLT(IRES))) = 0
000146          301   LET COST(IRES) = COSTD(TYPE(IRES))*TVEC(IRES)
000147                LET PRIOR(IRES) = PRI(ICNN)
000148                STORE CAS(ICNN) IN ACASE(IFLT(IRES))
000149                STORE ICNN IN FITON(IFLT(IRES))
000150                CALL SVQUE(ICNN,IRES)
000151                RETURN
000152        C
000153        C       FLG(ICNN) = 3; A SEARCH NEED IS BEING REMOVED FROM CQUE TO BE
000154        C       SERVED.  ATTRIBUTES ARE UPDATED AND THE ENDOGENOUS EVENT ARSCH IS
000155        C       CREATED AND CAUSED.
000156        C
000157          400   LET X = XC(ESAC(ICNN))-XS(STN(IRES))
000158                LET Y = YC(ESAC(ICNN))-YS(STN(IRES))
000159                LET D = SQRT(X*X+Y*Y)
000160                IF (SMELL(ESAC(ICNN)))GR(SLIM(TYPE(IRES))), GO TO 401
000161                LET XY = D/SOA1(TYPE(IRES))
000162                GO TO 402
000163          401   LET XY = D/SOA2(TYPE(IRES))
000164          402   IF (END(TYPE(IRES)))LE(2.0*XY), GO TO 999
000165          450   LET X = AINT(TIME) + RISE
000166                IF (TIME)LE(X), GO TO 500
000167                LET X = AINT(TIME) + SET
000168                IF (TIME)GR(X), GO TO 500
000169                IF (TVEC(IRES))LS(X-TIME), GO TO 500
000170                LET X = AINT(TIME) + RISE + 1.0
```

```
000171          IF (TIME+TVEC(IRES))GE(X), GO TO 500
000172          IF (VOP(TYPE(IRES)))EQ(1), GO TO 999
000173     500  IF (IVAR)EQ(0), GO TO 501
000174          LET IVAR = 0
000175          LET IWAIT(ACASE(IFLT(IRES))) = 0
000176          LET KRES(FITON(IFLT(IRES))) = 0
000177     501  LET CNRES(ESAC(ICNN)) = CNRES(ESAC(ICNN)) + 1
000178          LET TST = SM(ICNN)/SOA3(TYPE(IRES))
000179          LET COST(IRES) = COSTD(TYPE(IRES))*(TVEC(IRES)+TST)
000180          LET COSTC(ESAC(ICNN)) = COSTC(ESAC(ICNN)) + COST(IRES)
000181          LET RSRC(ICNN) = IRES
000182          LET I = SIGNL(ICNN)
000183          LET SIGNL(ICNN) = 2
000184          LET JIND = 2
000185          LET KIND = 2
000186          DO TO 506, FOR EACH J IN SRHS(ESAC(ICNN))
000187          IF (SIGNL(J))EQ(0), LET JIND = 1
000188          IF (SIGNL(J))EQ(1), LET KIND = 1
000189     506  LOOP
000190          IF (I)EQ(1), GO TO 507
000191          IF (JIND)EQ(1), GO TO 520
000192          IF (KIND)EQ(1), GO TO 520
000193          LET TQUE(ESAC(ICNN)) = TQUE(ESAC(ICNN))+(TIME-TINQ(ESAC(ICNN)))
000194          GO TO 520
000195     507  IF (KIND)EQ(1), GO TO 520
000196          IF (JIND)EQ(2), GO TO 510
000197          LET TINT(ESAC(ICNN)) = TINT(ESAC(ICNN))+(TIME-STINQ(ESAC(ICNN)))
000198          GO TO 520
000199     510  LET TQUE(ESAC(ICNN)) = TQUE(ESAC(ICNN))+(TIME-TINQ(ESAC(ICNN)))
000200          LET TINT(ESAC(ICNN)) = TINT(ESAC(ICNN))+(TIME-STINQ(ESAC(ICNN)))
000201     520  LET SASG(ICNN) = 1
000202          LET IB(IRES) = 1
000203          LET NCASE(IRES) = NCASE(IRES) + 1
000204          LET PRIOR(IRES) = PRI(ICNN)
000205          STORE ESAC(ICNN) IN ACASE(IFLT(IRES))
000206          LET DEP(IFLT(IRES)) = TIME
000207          STORE ICNN IN FITON(IFLT(IRES))
000208          LET XDEST(IFLT(IRES)) = XC(ESAC(ICNN))
000209          LET YDEST(IFLT(IRES)) = YC(ESAC(ICNN))
000210          CREATE ARSCH CALLED IDEV(IFLT(IRES))
000211          LET LNOTE(IDEV(IFLT(IRES))) = ICNN
000212          LET LRES(IDEV(IFLT(IRES))) = IRES
000213          LET II = 0
000214          DO TO 1, FOR EACH LL IN SRHS(ESAC(ICNN))
000215          IF (SFLAG(LL))EQ(1), LET II = 1
000216        1 LOOP
000217          IF (II)EQ(1), GO TO 2
000218          LET SFLAG(ICNN) = 1
000219          LET LFLG(IDEV(IFLT(IRES))) = 1
000220          GO TO 3
000221        2 LET LFLG(IDEV(IFLT(IRES))) = 2
000222        3 LET MFLG(IFLT(IRES)) = 3
000223          LET ROS(IFLT(IRES)) = TIME + TVEC(IRES)
000224          CAUSE ARSCH CALLED IDEV(IFLT(IRES)) AT TIME + TVEC(IRES)
000225          LET NEEDS(STN(IRES)) = NEEDS(STN(IRES)) + 1
000226          REMOVE ICNN FROM CQUE
000227          RETURN
000228     999  LOOP
```

```
000229          C
000230          C       RESOURCE CANNOT SERVE ANY CASE IN QUEUE; GO TO 15 SO THAT RESOURC:
000231          C       CAN BE VECTORED TO HOME STATION.
000232          C
000233                  IF (IVAR)EQ(C), GO TO 1
000234                  RETURN
000235                  END
```

```
000001                ENDOGENOUS EVENT STNBY
000002          C
000003          C     ENDOGENOUS EVENT STNBY OCCURS WHEN A STAND-BY CREW ARRIVES AT A
000004          C     STATION.  THE QUEUE IS EXAMINED.  IF THERE IS NO CASE TO SERVICE,
000005          C     THE STAND-BY CREW IS NOT ADDED TO THE AVAILABLE CREWS AT THE STA.
000006          C
000007                STORE STAT(STNBY) IN ISTA
000008                DESTROY STNBY
000009                IF CQUE IS EMPTY, GO TO 510
000010                DO TO 100, FOR EACH IJK IN CQUE
000011                IF (FLG(IJK))EQ(1), LET CASE = IJK
000012                IF (FLG(IJK))EQ(2), LET CASE = CAS(IJK)
000013                IF (FLG(IJK))EQ(3), LET CASE = ESAC(IJK)
000014                IF (SIS(CASE))GR(0), GO TO 210
000015                IF (NNN(CASE)+MMM(CASE))GR(1), GO TO 210
000016                GO TO (210,210,210,201,201,210), RAP
000017          201   IF (STATN(CASE))EQ(ISTA), GO TO 300
000018                DO TO 202, FOR I=(1)(NACS(STATN(CASE)))
000019                IF (ISTA)EQ(ACS(STATN(CASE),I)), GO TO 300
000020          202   LOOP
000021                DO TO 203, FOR I=(1)(NCUT(STATN(CASE)))
000022                IF (ISTA)EQ(CUT(STATN(CASE),I)), GO TO 300
000023          203   LOOP
000024                GO TO 100
000025          210   IF (STATN(CASE))EQ(ISTA), GO TO 300
000026                DO TO 211, FOR I=(1)(NADJS(STATN(CASE)))
000027                IF (ISTA)EQ(ADJS(STATN(CASE),I)), GO TO 300
000028          211   LOOP
000029                DO TO 212, FOR I=(1)(NACS(STATN(CASE)))
000030                IF (ISTA)EQ(ACS(STATN(CASE),I)), GO TO 300
000031          212   LOOP
000032                DO TO 213, FOR I=(1)(NCUT(STATN(CASE)))
000033                IF (ISTA)EQ(CUT(STATN(CASE),I)), GO TO 300
000034          213   LOOP
000035                DO TO 214, FOR I=(1)(NADJS(STATN(CASE)))
000036                DO TO 215, FOR K=(1)(NACS(ADJS(STATN(CASE),I)))
000037                IF (ISTA)EQ(ACS(ADJS(STATN(CASE),I),K)), GO TO 300
000038          215   LOOP
000039                DO TO 217, FOR K=(1)(NCUT(ADJS(STATN(CASE),I)))
000040                IF (ISTA)EQ(CUT(ADJS(STATN(CASE),I),K)), GO TO 300
000041          217   LOOP
000042          214   LOOP
000043                GO TO 100
000044          300   IF (FLG(IJK))EQ(1), CALL CRES(IJK,NEED(IJK),*KONTR,*II)
000045                IF (FLG(IJK))EQ(2), CALL CRES(CAS(IJK),NEED(IJK),*KONTR,*II)
000046                IF (FLG(IJK))EQ(3), CALL CRES(ESAC(IJK),18,*KONTR,*II)
000047                DO TO 400, FOR EACH IRS IN RQUE
000048                IF (STN(IRS))NE(ISTA), GO TO 400
000049                IF (EIAT(IRS))NE(0.0), GO TO 400
000050                IF (RAND*)GR(MR(TYPE(IRS))), GO TO 400
000051                IF (SQTAG(TYPE(IRS)))EQ(0), GO TO 350
000052                IF (PRI(CASE))GE(TPRI), GO TO 350
000053                LET II = 1
000054                DO, FOR EACH IR IN RQUE
```

```
000055                IF (SQTAG(TYPE(IR)))EQ(0), LET II = 0
000056                LOOP
000057                IF (II)EQ(1), GO TO 350
000058                IF (FLG(IJK))NE(3), GO TO 400
000059                DO TO 340, FOR EACH IR IN RQUE
000060                IF (SQTAG(TYPE(IR)))NE(0), GO TO 340
000061                LET X = XC(CASE) - XS(STN(IR))
000062                LET Y = YC(CASE) - YS(STN(IR))
000063                LET D = SQRT(X*X+Y*Y)
000064                IF (SWELL(CASE))GR(SLIM(TYPE(IR))), GO TO 338
000065                LET XY = D/SOA1(TYPE(IR))
000066                GO TO 339
000067           338  LET XY = D/SOA2(TYPE(IR))
000068           339  IF (END(TYPE(IR)))GR(2.0*XY), LET II = 1
000069           340  LOOP
000070                IF (II)EQ(1), GO TO 400
000071           350  LET BCREW(ISTA) = BCREW(ISTA) + 1
000072      C
000073      C         A SINGLE RESOURCE NEED, A MULTI-RESOURCE NEED OR A SEARCH NEED
000074      C         HAS BEEN LOCATED IN CQUE THAT CAN BE SERVED BY THE STANDBY CREW.
000075      C         A RESOURCE THAT CAN SERVE THE NEED HAS BEEN LOCATED IDLE AT THE
000076      C         STATION.  TVEC(IRS) IS CALCULATED.
000077      C
000078                LET X = XR(IRS)-XC(CASE)
000079                LET Y = YR(IRS)-YC(CASE)
000080                IF (SWELL(CASE))GR(SLIM(TYPE(IRS))), GO TO 600
000081                LET D = SQRT(X*X+Y*Y)
000082                LET TVEC(IRS) = D/SOA1(TYPE(IRS))
000083                GO TO 601
000084           600  LET D = SQRT(X*X+Y*Y)
000085                LET TVEC(IRS) = D/SOA2(TYPE(IRS))
000086           601  IF (FLG(IJK))EQ(2), GO TO 700
000087                IF (FLG(IJK))EQ(3), GO TO 800
000088      C
000089      C         FLG(IJK)=1; I.E., A SINGLE RESOURCE NEED IS BEING REMOVED FROM
000090      C         CQUE TO BE SERVED.
000091      C
000092                LET TQUE(CASE) = TQUE(CASE) + (TIME-TINQ(CASE))
000093                IF (SIGNL(CASE))EQ(1),LET TINT(CASE)=TINT(CASE)+(TIME-TINQ(CASE))
000094                LET COST(IRS) = COSTD(TYPE(IRS))*TVEC(IRS)
000095                LET PRIOR(IRS) = PRI(CASE)
000096                REMOVE CASE FROM CQUE
000097                CALL SERVE(CASE,0,IRS)
000098                RETURN
000099      C
000100      C         FLG(IJK)=2; A MULTI-RESOURCE NEED IS BEING REMOVED FROM CQUE TO
000101      C         BE SERVED.
000102      C
000103           700  LET COST(IRS) = COSTD(TYPE(IRS))*TVEC(IRS)
000104                LET PRIOR(IRS) = PRI(CASE)
000105                CALL SVQUE(IJK,IRS)
000106                RETURN
000107      C
000108      C         FLG(IJK) = 3; A SEARCH NEED IS BEING REMOVED FROM CQUE TO BE
000109      C         SERVED.  ATTRIBUTES ARE UPDATED AND THE ENDOGENOUS EVENT ARSCH
000110      C         IS CREATED AND CAUSED.
000111      C
000112           800  IF (END(TYPE(IRS)))GR(2.0*TVEC(IRS)), GO TO 850
```

```
000113          LET BCREW(ISTA) = BCREW(ISTA)-1
000114          GO TO 400
000115     850  IF (VOF(TYPE(IRS)))EQ(0), GO TO 860
000116     851  LET X = AINT(TIME) + RISE
000117          IF (TIME)LE(X), GO TO 860
000118          LET X = AINT(TIME) + SET
000119          IF (TIME)GR(X), GO TO 860
000120          IF (TVEC(IRS)+DLAY(TYPE(IRS)))LS(X-TIME), GO TO 860
000121          LET X = AINT(TIME) + RISE + 1.0
000122          IF (TIME+TVEC(IRS)+DLAY(TYPE(IRS)))GE(X), GO TO 860
000123          LET BCREW(ISTA) = BCREW(ISTA) - 1
000124          GO TO 400
000125     860  CREATE FLT CALLED IFLT(IRS)
000126          STORE CASE IN ACASE(IFLT(IRS))
000127          LET DEP(IFLT(IRS)) = 0.0
000128          STORE IJK IN FITON(IFLT(IRS))
000129          LET TFLT(IFLT(IRS)) = 0.0
000130          LET HCREW(IFLT(IRS)) = 0
000131          LET XDEST(IFLT(IRS)) = 0.0
000132          LET YDEST(IFLT(IRS)) = 0.0
000133          LET MFLG(IFLT(IRS)) = 12
000134          LET POS(IFLT(IRS)) = 0.0
000135          CREATE READY CALLED IDEV(IFLT(IRS))
000136          LET NREAD(IDEV(IFLT(IRS))) = IJK
000137          LET RREAD(IDEV(IFLT(IRS))) = IRS
000138          LET II = 0
000139          DO TO 1, FOR EACH LL IN SRHS(CASE)
000140          IF (SFLAG(LL))EQ(1), LET II = 1
000141        1 LOOP
000142          IF (II)EQ(1), GO TO 2
000143          LET SFLAG(IJK) = 1
000144          LET FREAD(IDEV(IFLT(IRS))) = 1
000145          GO TO 3
000146        2 LET FREAD(IDEV(IFLT(IRS))) = 2
000147        3 CAUSE READY CALLED IDEV(IFLT(IRS)) AT TIME + DLAY(TYPE(IRS))
000148          LET CNRES(CASE) = CNRES(CASE) + 1
000149          LET TST = SM(IJK)/SOA3(TYPE(IRS))
000150          LET COST(IRS) = COSTD(TYPE(IRS))*(TVEC(IRS)+TST)
000151          LET COSTC(CASE) = COSTC(CASE) + COST(IRS)
000152          LET RSRC(IJK) = IRS
000153          LET I = SIGNL(IJK)
000154          LET SIGNL(IJK) = 2
000155          LET JIND = 2
000156          LET KIND = 2
000157          DO TO 861, FOR EACH J IN SRHS(CASE)
000158          IF (SIGNL(J))EQ(0), LET JIND = 1
000159          IF (SIGNL(J))EQ(1), LET KIND = 1
000160     861  LOOP
000161          IF (I)EQ(1), GO TO 807
000162          IF (JIND)EQ(1), GO TO 820
000163          IF (KIND)EQ(1), GO TO 820
000164          LET TQUE(CASE) = TQUE(CASE) + (TIME-TINQ(CASE))
000165          GO TO 820
000166     807  IF (KIND)EQ(1), GO TO 820
000167          IF (JIND)EQ(2), GO TO 810
000168          LET TINT(CASE) = TINT(CASE) + (TIME-STINQ(CASE))
000169          GO TO 820
000170     810  LET TQUE(CASE) = TQUE(CASE) + (TIME-TINQ(CASE))
```

```
000171                LET TINT(CASE) = TINT(CASE) + (TIME-STIND(CASE))
000172          820   LET SASG(IJK) = 1
000173                LET FIAT(IRS) = 1.0
000174                LET IB(IRS) = 1
000175                LET NCASE(IRS) = NCASE(IRS) + 1
000176                LET PRIOR(IRS) = PRI(IJK)
000177                LET NEEDS(STN(IRS)) = NEEDS(STN(IRS)) + 1
000178                REMOVE IJK FROM QUE
000179                RETURN
000180          400   LOOP
000181          100   LOOP
000182          510   LET UNPRO(ISTA) = UNPRO(ISTA) + 1
000183                LET TUNPR = TUNPR + 1
000184                RETURN
000185                END
```

```
@ ELT SVQUE,1,71-520, 65944

000001                SUBROUTINE SVQUE(NOTIF,IRES)                                    130000
000002                LET CASE = CAS(NOTIF)                                           130010
000003                REMOVE NOTIF FROM CQUE                                          130020
000004                LET COMP(NOTIF) = 2                                             130030
000005                LET KRES(NOTIF) = IRES                                          130040
000006                LET N1 = 0                                                      130050
000007                LET N2 = 0                                                      130060
000008                DO TO 10, FOR EACH INOTF IN NSET(CASE)                          130070
000009                IF COMP(INOTF) NE 1, GO TO 10                                   130080
000010                LET N1 = N1 + 1                                                 130090
000011                IF SIGNL(INOTF) EQ 0, GO TO 10                                  130100
000012                LET N2 = N2 + 1                                                 130110
000013             10 LOOP                                                            130120
000014                IF N1 EQ 0, LET TQUE(CASE) = TQUE(CASE) + TIME-TINQ(CASE)       130130
000015                IF N2 GR 0, GO TO 20
000016                IF SIGNL(NOTIF) EQ 1, LET TINT(CASE) = TINT(CASE) + TIME -      130160
000017               .STINQ(CASE)                                                     130170
000018             20 LET SIGNL(NOTIF) = 2
000019                CALL SERVE(CASE,NOTIF,IRES)
000020                RETURN                                                          130190
000021                END OF SVQUE                                                    130200
```

```
      SUBROUTINE RECK(NOTIF)                                            120000
C     THIS SUBROUTINE CANCELS AND DESTROYS NOTIFS, ASSOCIATED ONSCN A D 120010
C     ARVSN ENDOGENOUS EVENTS, AND CASES.                               120020
C                                                                       120030
      LET NTIFC = CAS(NOTIF)                                            120060
    1 IF NSET(NTIFC) IS EMPTY, GO TO 20                                 120070
      REMOVE FIRST NOTIF FROM NSET(NTIFC)                               120080
      IF COMP(NOTIF) GE 3, GO TO 15                                     120120
      IF (COMP(NOTIF))EQ(1), GO TO 7                                    120110
      IF (COMP(NOTIF))EQ(2), GO TO 5                                    120100
    3 CANCEL NOTIF                                                      120140
      GO TO 9                                                           120150
    7 REMOVE NOTIF FROM CQUE                                            120160
      GO TO 9                                                           120170
    5 IF KRES(NOTIF) EQ 0, GO TO 15
      IF MFLG(IFLT(KRES(NOTIF))) EQ 1, GO TO 11
      IF (MFLG(IFLT(KRES(NOTIF))))EQ(2), GO TO 13                       120190
      IF MFLG(IFLT(KRES(NOTIF))) EQ 13, GO TO 6
      IF MFLG(IFLT(KRES(NOTIF))) EQ 11, GO TO 4
      GO TO 9                                                           120200
   11 CANCEL ARVSN CALLED IDEV(IFLT(KRES(NOTIF)))                       120210
      DESTROY ARVSN CALLED IDEV(IFLT(KRES(NOTIF)))                      120220
      GO TO 8                                                           120230
    6 CANCEL CHEKN CALLED IDEV(IFLT(KRES(NOTIF)))
      DESTROY CHEKN CALLED IDEV(IFLT(KRES(NOTIF)))
      GO TO 9
    4 CANCEL  DELAY CALLED IDEV(IFLT(KRES(NOTIF)))
      DESTROY DELAY CALLED IDEV(IFLT(KRES(NOTIF)))
      GO TO 9
   13 CANCEL ONSCN CALLED IDEV(IFLT(KRES(NOTIF)))                       120240
      DESTROY ONSCN CALLED IDEV(IFLT(KRES(NOTIF)))                      120250
      IF NUMBR(NOTIF) EQ 0, GO TO 9                                     120260
    8 LET I = KRES(NOTIF)                                               120270
      LET X = XR(I) - XDEST(IFLT(I))                                    120290
      LET Y = YR(I) - YDEST(IFLT(I))                                    120300
      LET D = SQRT(X*X+Y*Y)                                             120310
      IF NEED(NOTIF) EQ 17, GO TO 30                                    120320
      IF NUMBR(NOTIF) NE 0, GO TO 35                                    120330
      IF SWELL(NTIFC) GR SLIM(TYPE(I)), GO TO 31                        120340
   30 LET SPEED = SOA1(TYPE(I))                                         120350
      GO TO 39                                                          120360
   31 LET SPEED = SOA2(TYPE(I))                                         120370
      GO TO 39                                                          120380
   35 IF L(NTIFC) GR 26, GO TO 37
      LET SPEED = TSP1                                                  120400
      GO TO 39                                                          120410
   37 LET SPEED = TSP2                                                  120420
   39 LET C = (TIME-DEP(IFLT(I)))*SPEED                                 120430
      IF C LS 0., LET C = 0.
      LET XR(I) = XR(I) - (C*X)/D                                       120440
      LET YR(I) = YR(I) -(C*Y)/D                                        120450
      LET XDEST(IFLT(I)) = 0.                                           120460
      LET YDEST(IFLT(I)) = 0.                                           120470
      LET DEP(IFLT(I)) = 0.                                             120480
```

```
000055          9 LET IRES = KRES(NOTIF)                                  120500
000056            IF(IRES)EQ(0),GO TO 15                                  120520
000057            LET IB(IRES) = 1                                        120490
000058            LET MFLG(IFLT(IRES)) = 9
000059            LET IVAR = 0
000060            CALL EXQ(IRES,**IVAR)
000061         15 DESTROY NOTIF                                           120540
000062            GO TO 1                                                 120550
000063         20 FILE NTIFC IN EXCS                                      120560
000064            RETURN                                                  120570
000065            END                                                     120580
```

```
@ ELT ENDSIM,1,7,0520, 65944

 EXOGENOUS EVENT ENDSIM                                                 15000
DIMENSION IBUFF(15),FBUFF(10)
LET TOTME(PSHFT) = TOTME(PSHFT) + TIME - TLAST                          15010
DO TO 10, FOR EACH RES IRS                                              15020
IF EIAT(IRS) EQ 0., GO TO 6                                             15030
IF TFLT(IFLT(IRS)) EQ 0.0, GO TO 6
IF TFLT(IFLT(IRS)) GT TLAST, GO TO 5                                    15040
LET TUTIL(IRS) = TUTIL(IRS) + TIME - TLAST                              15050
GO TO 6                                                                 15060
 5 LET TUTIL(IRS) = TUTIL(IRS) + TIME - TFLT(IFLT(IRS))                 15070
 6 LET UTIL(IRS) = UTIL(IRS) + TUTIL(IRS)                               15080
LET USHF(STN(IRS),PSHFT) = USHF(STN(IRS),PSHFT) + TUTIL(IRS)            15090
10 LOOP                                                                 15100
LET IJK = 0
DO, FOR EACH RES I, WITH (SQTAG(TYPE(I)))EQ(0)
LET BUTIL = BUTIL + UTIL(I)
LET IJK = IJK + 1
LOOP
LET BUTIL = BUTIL*100./(TIME*FLOAT(IJK))
LET IJK = 0
DO, FOR EACH RES I,WITH (SQTAG(TYPE(I)))EQ(1)
LET CUTIL = CUTIL + UTIL(I)
LET IJK = IJK + 1
LOOP
LET CUTIL = CUTIL*100.0/(TIME*FLOAT(IJK))
LET IJK = 0
DO, FOR EACH RES I, WITH (SQTAG(TYPE(I)))EQ(3)
LET AIRU = AIRU + UTIL(I)
LET IJK = IJK + 1
LOOP
LET AIRU = AIRU*100./(TIME*FLOAT(IJK))
LET R = MNTMT/MEAND
DO, FOR EACH STA I
LET CS(GRP(I)) = CS(GRP(I)) + NCAS(I)
LET NDS(GRP(I)) = NDS(GRP(I)) + NEEDS(I)
LET FL1(GRP(I)) = FL1(GRP(I)) + FAIL1(I)
LET FL2(GRP(I)) = FL2(GRP(I)) + FAIL2(I)
LET FL3(GRP(I)) = FL3(GRP(I)) + FAIL3(I)
LET INTRP(GRP(I)) = INTRP(GRP(I)) + NINTR(I)
LET NOSB(GRP(I)) = NOSB(GRP(I)) + NSTRY(I)
LET NONPR(GRP(I)) = NONPR(GRP(I)) + UNPRO(I)
DO, FOR EACH WENDS K
LET USEAV(GRP(I)) = USEAV(GRP(I)) + USHF(I,K)
LOOP
LOOP
DO TO 200, FOR EACH GROUP I
 LET NUM = 0
 DO TO 300, FOR EACH STA II, WITH (GRP(II))EQ(I)
 DO TO 330, FOR EACH RST III
 LET NUM = NUM + REST(II,III)
330 LOOP
300 LOOP
 IF NUM EQ 0,/GO TO 200
 IF (USEAV(I))LE(0.0), GO TO 1
```

```
000055          LET USEAV(I) = (USEAV(I)*100.0)/(TIME*FLOAT(NUM))
000056        1 IF CS(I) LE 0, GO TO 200
000057          LET X = FLOAT(CS(I))
000058          LET TWAVG(I) = TWAVG(I)*24./X
000059          LET TVAVG(I) = TVAVG(I)*24./X
000060          LET TMTAV(I) = TMTAV(I)*24./X
000061          LET AVDRT(I) = (AVDRT(I) * 24.0 * R)/X
000062      200 LOOP
000063          DO, FOR I=(1)(2)
000064          LET X = FLOAT(CNTR(I))*FLOAT(NRES)
000065          LET MEEN(I) = MEEN(I)/X
000066          LET X = X * FLOAT(NRES)
000067          LET STDEV(I) = ABS((STDEV(I)/X)-(MEEN(I)*MEEN(I)))
000068          LET STDEV(I) = SQRT(STDEV(I)) * 100.0
000069          LET MEEN(I) = MEEN(I) * 100.0
000070         LOOP
000071          DO, FOR I=(1)(10)
000072          LET K = I + 2
000073          LET MEEN(K) = MEEN(K)/FLOAT(CNTR(K))
000074          LET STDEV(K)=ABS((STDEV(K)/FLOAT(CNTR(K)))-(MEEN(K)*MEEN(K)))
000075          LET STDEV(K) = SQRT(STDEV(K)) * 24.0
000076          LET MEEN(K) = MEEN(K) * 24.0
000077          LOOP
000078       20 DO TO 40, FOR EACH WEWDS J                                    15110
000079          DO TO 30, FOR EACH STA I                                      15120
000080          LET AVUS(J) = AVUS(J) + USHF(I,J)                             15130
000081       30 LOOP                                                          15140
000082          LET AVUS(J) = AVUS(J)*100./(TOTME(J)*FLOAT(NRES))             15160
000083       40 LOOP                                                          15170
000084          DO TO 50, FOR EACH RES I                                      15190
000085          LET AUT(TYPE(I)) = AUT(TYPE(I)) + UTIL(I)                     15200
000086       50 LOOP                                                          15210
000087          DO TO 70, FOR EACH RST I                                      15220
000088          LET NUM = 0                                                   15230
000089          DO TO 60, FOR EACH STA J                                      15240
000090          LET NUM = NUM + REST(J,I)                                     15250
000091       60 LOOP                                                          15260
000092          IF NUM LE 0, GO TO 70                                         15270
000093          LET AUT(I) = AUT(I)*100./(TIME*FLOAT(NUM))                    15280
000094          IF (SOTAG(I))EQ(2), LET C130U = AUT(I)
000095       70 LOOP                                                          15290
000096          LET NUMB = 0                                                  15298
000097          LET MEANW = 0.                                                15299
000098          LET KOUNT = 0
000099          LET LIMIT = 0
000100          DO TO 100, FOR EACH STA J                                     15300
000101          LET SNEED = SNEED + NEEDS(J)
000102          LET NBRFA = NBRFA + FAIL1(J)                                  15310
000103          LET NBRFB = NBRFB + FAIL2(J)                                  15320
000104          LET NBRFC = NBRFC + FAIL3(J)                                  15330
000105          LET MEANW = MEANW + AVGTW(J)                                  15302
000106          LET LIMIT = LIMIT + NMBRO(J)
000107          LET NUMB = NUMB + NCAS(J)                                     15304
000108          IF NCAS(J) LE 0, GO TO 70
000109          LET KOUNT = KOUNT + NCAS(J)
000110          LET AVGTW(J) = AVGTW(J)*24./(FLOAT(NCAS(J)))
000111          IF FAIL3(J) GR 0, LET CFTT(J)=TWTOL(J)*24.0/(FLOAT(FAIL3(J)))
000112          LET TWTOL(J) = TWTOL(J)*24./(FLOAT(NCAS(J)))
```

```
000113          LET VCTR(J) = VCTR(J)*24./(FLOAT(NCAS(J)))
000114          LET DMRI(J) = (DMRT(J)*24.0*R)/(FLOAT(NCAS(J)))
000115       70 LET NUM = 0
000116          DO TO 80, FOR EACH RST I                                        15350
000117          LET NUM = NUM + REST(J,I)                                       15360
000118       80 LOOP                                                            15370
000119          DO TO 90, FOR EACH MENDS I                                      15380
000120          LET USE(J) = USE(J) + USHF(J,I)                                 15390
000121          IF TOTME(I) LE 0., GO TO 90
000122          IF NUM LE 0, GO TO 90
000123          LET USHF(J,I) = USHF(J,I)*100./(TOTME(I)*FLOAT(NUM))            15400
000124       90 LOOP                                                            15410
000125          IF NUM LE 0, GO TO 100
000126          LET USE(J) = USE(J)*100./(TIME*FLOAT(NUM))                      15420
000127          LET AVUTO = AVUTO + USE(J)*FLOAT(NUM)
000128      100 LOOP                                                            15430
000129          LET AVUTO = AVUTO/FLOAT(NRES)
000130          LET MEANW = MEANW*24./FLOAT(NUMB)
000131          IF NBRFC GR 0, LET MCFTT = MNTMT*24.0/FLOAT(NBRFC)
000132          LET MNTMT = MNTMT*24.0/FLOAT(NUMB)
000133          LET MEANV = MEANV*24.0/FLOAT(NUMB)
000134          LET MEAND = (MEAND*24.0*R)/FLOAT(NUMB)
000135          DO TO 110, FOR EACH RES I                                       15440
000136          LET UTIL(I) = UTIL(I)*100./TIME                                 15450
000137      110 LOOP                                                            15460
000138          CALL DRIVE(R)
000139          IF STAPE EQ 0, STOP
000140          IF EXCS IS EMPTY, GO TO 875
000141          DO TO 800, FOR I = (1)(15)
000142          LET IBUFF(I) = 0
000143          IF I GR 10, GO TO 800
000144          LET FBUFF(I) = 0.
000145      800 LOOP
000146          LET NUMBB = NBRCO + 1
000147          DO TO 850, FOR EACH CASE IN EXCS
000148          WRITE ON TAPE STAPE, NUMBB,OPFAC(CASE),NOCAS(CASE),IDLOC(CASE),
000149         *OCCUR(CASE),SOX(CASE),FPRI(CASE),MMM(CASE),NNN(CASE),GAMMA(CASE),
000150         *NEED(CASE),ATR(CASE),OFSHR(CASE),VIS(CASE),WIND(CASE),SWELL(CASE)
000151         *L(CASE),POB(CASE),S1S(CASE),S2S(CASE),TSM(CASE),OST(CASE)
000152          FORMAT (3I5,I3,D3.4,I5,I1,2I2,D3.2,I2,I5,D4.2,6I5,I2,D5.0,D1.4)
000153          WRITE ON TAPE STAPE, UTYPE(CASE),VALUE(CASE),XCX(CASE),YCY(CASE),
000154         *XC(CASE),YC(CASE),STATN(CASE),CNRES(CASE),RESA(CASE),PRI(CASE),
000155         *REA(CASE),COSTC(CASE),ITOL(CASE),NOINT(CASE),NQUE(CASE),
000156         *TINT(CASE),TQUE(CASE),TQUEI(CASE),TSVC(CASE),TWAIT(CASE)
000157          FORMAT (I5,I10,4D5.2,I5,I2,I3,2I1,D7.2,I1,2I2,5D3.4)
000158          WRITE ON TAPE STAPE, IBUFF(I),FBUFF(I),FBUFF(I+1),IBUFF(I+1), FOR
000159         * I = (1)(10)(2)
000160          FORMAT 5(I2,D1.4,D1.2,I3)
000161          WRITE ON TAPE STAPE, IBUFF(I), FOR I=(11)(15)
000162          FORMAT 5(I3)
000163          LET NUMBB = NUMBB + 1
000164      850 LOOP
000165      875 IF STAPE EQ 6, STOP
000166          ENDFILE STAPE
000167           STOP                                                           15610
000168           END                                                            15620
```

```
@ ELT DRIVE,1,710520, 65945

000001                SUBROUTINE DRIVE(G)
000002                CALL JUMPER
000003          X      NOP 0,6969
000004                CALL SARSIN
000005                WRITE ON TAPE 6, P
000006                FORMAT(/' NOTE:  1. SCALING FACTOR FOR NORMLZD DEMERIT (12TH COLU
000007               *N ABOVE)    =  ',D1.3,/,S8,'2. COEFFICIENTS USED IN THE CALCULATION
000008               * OF CASE DEMERIT:  1,2,3')
000009                CALL GRPRES
000010                CALL HEADER
000011                DO TO 600, FOR EACH STA N
000012                IF (XS(N))NE(0.0), GO TO 500
000013                IF (YS(N))NE(0.0), GO TO 500
000014                GO TO 600
000015            500 CALL TITLE(N)
000016                DO TO 601, FOR EACH RST J, WITH (REST(N,J))GR(0)
000017                CALL RESULT(N,J)
000018            601 LOOP
000019            600 LOOP
000020                CALL HEAD
000021                DO TO 602, FOR EACH CASE IN EXCS
000022                CALL EXCASE(CASE)
000023            602 LOOP
000024                CALL DSTRIB
000025                IF CQUE IS EMPTY, GO TO 730
000026                WRITE ON TAPE 6, TIME
000027                FORMAT (*1',S6,'THE FOLLOWING CASES WERE IN THE QUEUE WHEN THE SIM
000028               *ULATION ENDED AT',M4.2.2///S9,'TIME OF',S48,'QUEUED',S10,'*',S6,
000029               *'***'/'  CASE  OCCURRENCE  STATION  NNN  MMM  S1S  S2S    XC      YC
000030               *     AT TIME   REASON   TYPE')
000031                DO TO 720, FOR EACH ICNN IN CQUE
000032                IF FLG(ICNN) NE 0, GO TO (702,704,706),FLG(ICNN)
000033            702 LET CASE = ICNN
000034                GO TO 710
000035            704 LET CASE = CAS(ICNN)
000036                GO TO 710
000037            706 LET CASE = ESAC(ICNN)
000038            710 WRITE ON TAPE 6, NOCAS(CASE),OCCUR(CASE),STATN(CASE),NNN(CASE),
000039               *MMM(CASE),S1S(CASE),S2S(CASE),XC(CASE),YC(CASE),TINQ(CASE),
000040               *SIGNL(ICNN),FLG(ICNN)
000041                FORMAT (I6,S3,M3.2.2,2I7,3I5,2D5.1,S2,M3.2.2,I6,I8)
000042            720 LOOP
000043                LET I1 = 1
000044                WRITE ON TAPE 6, I1
000045                FORMAT (*0 *  0 = NO AVAILABLE RESOURCES,',I2,' = INTERRUPT'/'0.*
000046               * 1 = SINGLE RESOURCE CASE, 2 = NEED OR TOW NOTIF, 3 = SEARCH NOTE,
000047               *)
000048                GO TO 740
000049            730 WRITE ON TAPE 6, TIME
000050                FORMAT (*1',S9,'THERE WERE NO CASES IN THE QUEUE WHEN THE SIMULATI
000051               *ON ENDED AT',M4.2.2)
000052            740 LET I1 = 0
000053                DO TO 790, FOR EACH RES IRS
000054                IF IB(IRS) EQ 0, GO TO 790
```

```
000055          IF I1 EQ 1, GO TO 750
000056          WRITE ON TAPE 6, TIME
000057          FORMAT (*1*,S5,*THE FOLLOWING RESOURCES WERE BUSY WHEN THE SIMULAT
000058         *ION ENDED AT*,M4.2.2///S10,*RESOURCE*,S43,*CASE*/* RESOURCE  STATI
000059         *ON      XR      YR    IB    CASE       OCCUR     STATION  NNN  MMM  SIS
000060         * S2S      XC       YC      TWAIT*)
000061          LET I1 = 1
000062      750 LET CASE = ACASE(IFLT(IRS))
000063          WRITE ON TAPE 6, IRS,STN(IRS),XR(IRS),YR(IRS),IB(IRS),OPFAC(CASE),
000064         *NOCAS(CASE),OCCUR(CASE),STATN(CASE),NNN(CASE),MMM(CASE),SIS(CASE),
000065         *S2S(CASE),XC(CASE),YC(CASE),TWAIT(CASE)
000066          FORMAT (I7,I9,S3,2D5.1,2I4,I5,M5.2.2,2I7,3I5,2D6.1,M5.2.2)
000067      790 LOOP
000068          IF I1 EQ 1, RETURN
000069          WRITE ON TAPE 6, TIME
000070          FORMAT (/////S9,*THERE WERE NO BUSY RESOURCES WHEN THE SIMULATION
000071         *ENDED AT*,M4.2.2)
000072          RETURN
000073          END
```

```
D ELT SARSIM,1,7)0520, 65945

000001                          REPORT SARSIM
000002          X                  3. OPSIM
000003          X                     RAP     = **
000004           X                        RAP
000005          X                     TOL(1) = **.** DAYS      TOLS(1) = **.** DAYS
000006           X      TOL(1)       TOLS(1)
000007          X                     TOL(2) = **.** DAYS      TOLS(2) = **.** DAYS
000008           X      TOL(2)       TOLS(2)
000009          X                     TOL(3) = **.** DAYS      TOLS(3) = **.** DAYS
000010           X      TOL(3)       TOLS(3)
000011          X                     TOL(4) = **.** DAYS      TOLS(4) = **.** DAYS
000012           X      TOL(4)       TOLS(4)
000013          X                     TOL(5) = **.** DAYS      TOLS(5) = **.** DAYS
000014           X      TOL(5)       TOLS(5)
000015          X                     IDELT   = **
000016           X                        IDELT
000017          X                     KKK     = **.**
000018           X                       KKK
000019          X                     PRDN    = **.**
000020           X                       PRDN
000021          X                     PRUP    = **.**
000022           X                       PRUP
000023          X                     HO      = **.** NAUTICAL MILES
000024           X                       HO
000025          X                                                          II.  DISTRIB
000026          X      A.  CASE SUMMARY
000027          X            TOTAL NUMBER OF CASES THAT OCCURRED =        *****
000028           X                                                        NBRCS
000029          X            TOTAL NUMBER OF CASES THAT WERE COMPLETED =  *****
000030           X                                                        NBRCO
000031          X            TOTAL NUMBER OF CASES WITH FAILURE A =       *****
000032           X                                                        NBRFA
000033          X            TOTAL NUMBER OF CASES WITH FAILURE B =       *****
000034           X                                                        NBRFB
000035          X            TOTAL NUMBER OF CASES WITH FAILURE C =       *****
000036           X                                                        NBRFC
000037          X            TOTAL SIMULATED TIME (DAYS) =                *****.
000038           X                                                         TIME
000039          X      B.  RESOURCE SUMMARY
000040          X            AVERAGE UTILIZATION OVERALL =**.**%
000041           X                                  AVUTO
000042          X            AVERAGE UTILIZATION BY SHIFTS:
000043          X                                 SHIFT *  -  **.*.%
000044           X                                   JJ     AVUS(JJ)
000045            X           FOR EACH AENDS JJ
000046          X            AVERAGE UTILIZATION BY RESOURCE TYPES:
000047          X                                 RESOURCE TYPE *  -  **.**%
000048           X                                          JJ    AUT(JJ)
000049            X           FOR EACH RST JJ
000050          X            COMBINED UTILIZATION OF BOATS            -   **.**%
000051           X                                                        BUTIL
000052          X            COMBINED UTILIZATION OF CUTTERS          -   **.**%
000053           X                                                        CUTIL
000054          X            COMBINED UTILIZATION OF C130             -   **.**%
```

```
000055          X                                                          CIBND
000056         X          COMBINED UTILIZATION OF OTHER AIRCRAFT    -    **.***8
000057          X                                                          AIRD
000058         X      C.  STATION SUMMARY
000059         X          TOTAL NUMBER OF TIMES A STANDBY WAS CALLED (ALL STATIONS) =
000060          X
000061         X          TOTAL NUMBER OF TIMES A STANDBY WAS CALLED BUT NOT USED (AL
000062          X
000063         X                                                      III.  STATION
000064         X           NUMBER NUMBER FAILURE FAILURE FAILURE NO.    TOTAL    AVERA
000065         X             OF     OF     TYPE    TYPE    TYPE  OF  INTERRUPTED  TVEC
000066         X            CASES  NEEDS    A       B       C     Q     NEEDS    (HOUR
000067         X      STA **  ***    ***    **      **      **    **      *       *.**
000068          X    I,NCAS(I),NEEDS(I),FAIL1(I),FAIL2(I),FAIL3(I),NMBRQ(I),NINTR(I),VC
000069           X                FOR EACH STA I
000070         X      TOTAL/AVG ****  ****   **      **      **    **     **       *.**
000071          X            KOUNT  SNEED  NBRFA  NBRFB   NBRFC LIMIT  TOTIN     MEANV
000072                 END
000073
000074                                                                            X
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097         T STATISTICS                                                      2X
000098                                                                           1
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109                                                                           2
000110
000111                                                                           1
000112
```

```
000113
000114
000115
000116
000117
000118                                                                  1 XX
000119
000120
000121
000122                                                                  1
000123                                                                       X
000124                                                                  1
000125                                                                       X
000126                                                                  1
000127                                                                       X
000128                                                                  12
000129
000130                                                                  11
000131          *****                                                   1
000132          TOSBY
000133           STATIONS) = *.***                                      1
000134          TUNPR
000135          RESPONSE                                                12X
000136          GE AVERAGE AVG. CFAIL AVG. POS. NORMLZD   STANDBYS     AVERAGE
000137              TWAIT   TWAIT-TOL TWAIT-TOL DEMERIT CALLS/UNPRO. UTILIZATION
000138          S) (HOURS)   (HOURS)   (HOURS)  (HOURS)                         1
000139              *.**      *.**      *.**      *.**       **/ *       **.**%  1
000140          TR(I),AVGTW(I),CFTT(I),TWTOL(I),DMRT(I),NSTBY(I),UNPRO(I),USE(I)
000141
000142              *.**      *.**      *.**      *.**       **/  *      **.**%
000143             MEANW     MCFTT     MNTMT     MEAND  TOSBY TUNPR     AVUTO
000144                END
```

```
@ ELT GRPRES,1,710520, 65945

000001                REPORT GRPRES
000002          X                                                                  G
000003          X              NUMBER  NUMBER  FAILURE  FAILURE  FAILURE     TOTAL     AV
000004          X                OF      OF      TYPE     TYPE     TYPE   INTERRUPTED   T
000005          X               CASES   NEEDS     A        B        C        NEEDS     (H
000006          X     GROUP **  ***     ***      **       **       **         *         *
000007           X    I,CS(I),NDS(I),FL1(I),FL2(I),FL3(I),INTRP(I),TVAVG(I),TWAVG(I),TMT
000008             X  FOR EACH GROUP I
000009                END
000010
000011          OUP RESPONSE                                                      12
000012          ERAGE  AVERAGE AVG. POS. NORMALIZED   TIMES   UNPRO    AVERAGE
000013          VEC     TWAIT  TWAIT-TOL   DEMERIT   STANDBY STANDBY UTILIZATION
000014          OURS)  (HOURS)  (HOURS)    (HOURS)    CALLED  CALLS                1
000015          .**     *.**     *.**       *.**       **      **      **.**%       1
000016          AV(I),AVDRT(I),NOSB(I),NONPR(I),USEAV(I)
000017
000018                END
```

```
@ ELT HEADER,1,/10520, 65045

000001                                  REPORT HEADER
000002            X                                                                    IV .  RESOURCE UTI
000003                  END
000004
000005            LIZATION (PERCENT)                                                          2
000006                  END
```

```
R ELT TITLE,1,710520, 65745

000001                          REPORT TITLE(1,
000002                 8        FOR EACH VEVDS J
000003            X                                SHIFT1       SHIFT2       SHIFT3       SHIFT4
000004            X      STATN ***                 **.**8       **.**8       **.**8       **.**8
000005             X            N                  5(USHF( ,J))
000006                   END
000007                                                                                        X
000008
000009                   SHIFT5       SHIFT6       SHIFT7       SHIFT8                       1
000010                   **.**8       **.**8       **.**8       **.**8
000011
000012                   END
```

```
@ ELT RESULT,1,710520, 65945

000001                  REPORT RESULT(NN,JJ)
000002          X        RST ***
000003           X                    JJ
000004          X          RES *** (ASSIGNED TO **** NEEDS) = **,***
000005           X              K              NCASE(K)         UTIL(K)
000006            X     FOR EACH RES K, WITH (TYPE(K))EQ(JJ), AND (STN(K))EQ(NN)
000007                 END
000008
000009                                                                                  1   X
000010
000011
000012
000013
000014                 END
```

```
@ ELT HEAD,I,710520, 65948

000001                          REPORT HEAD
000002          X                                                   V.  EXCEPTIONAL C
000003                 END
000004
000005          ASES                                                          3
000006                 END
```

```
@ ELT EXCASE,1,710520, 65045

      SUBROUTINE EXCASE(CASE)
      WRITE ON TAPE 6,NOCAS(CASE),STATN(CASE),S2S(CASE),L(CASE),
     * POB(CASE),UTYPE(CASE),OFSHR(CASE),OST(CASE),XC(CASE),TSM(CASE)
      FORMAT (S1,'CASE NO.',I6,'  STATN',I4,'    S2S',I3,'  LENGTH',I5,
     1 '    POB',I5,'  UTYPE',I5,'  OFSHR',D5.2,'    OST',D3.2,'   XC',
     2 D5.1,'    TSM',D4.1)
      WRITE ON TAPE 6,NNN(CASE),FPRI(CASE),ITOL(CASE),SWELL(CASE),
     * NOINT(CASE),GAMMA(CASE),TINT(CASE),XCX(CASE),TSVC(CASE)
      FORMAT (S19,'NNN',I4,'   FPRI',I3,'    ITOL',I5,'  SWELL',I5,
     1 '  NOINT',I5,'  GAMMA',D5.2,'   TINT',D3.2,'  XCX',D5.1,
     2 '   TSVC',D4.1)
      WRITE ON TAPE 6,MMM(CASE),PRI(CASE),NEED(CASE),VIS(CASE),
     * NQUE(CASE),COSTC(CASE),TQUE(CASE),YC(CASE),TWAIT(CASE)
      FORMAT (S19,'MMM',I4,'    PRI',I3,'    NEED',I5,'    VIS',I5,
     1 '   NQUE',I5,'   COST',D5.2,'   TQUE',D3.2,'   YC',D5.1,
     2 '  TWAIT',D4.1)
      WRITE ON TAPE 6,SIS(CASE),IDLOC(CASE),AIR(CASE),WIND(CASE),
     * CNRES(CASE),OCCUR(CASE),TQUE1(CASE),YCY(CASE)
      FORMAT (S19,'SIS',I4,'  IDLOC',I3,'     AIR',I5,'   WIND',I5,
     1 '  CNRES',I5,'  OCCUR',D5.2,'  TQUE1',D3.2,'  YCY',D5.1,//)
      RETURN
      END
```

```
@ ELT DSTRIP,1,710520, 65945

000001                REPORT DSTRIB
000002          X                                                          VI.  DIST
000003          X        A.  SYSTEM
000004          X                                                   -----------------
000005          X                                                   MEAN  STD. DEV.  D
000006          X            DAILY UTILIZATION OVERALL (WEEKDAYS)  -   *.**     *.**
000007           X     MEEN(1),STDEV(1),CATG1(1),CATG2(1),CATG3(1),CATG4(1),CATG5(1),CATG
000008          X            DAILY UTILIZATION OVERALL (WEEKENDS)  -   *.**     *.**
000009           X     MEEN(2),STDEV(2),CATG1(2),CATG2(2),CATG3(2),CATG4(2),CATG5(2),CATG
000010          X        B.  CASES
000011          X                                                   -----------------
000012          X                                                   MEAN  STD. DEV.  D
000013          X            TWAIT  (ALL CASES)       WEEKDAYS      -   *.**     *.**
000014           X     MEEN(3),STDEV(3),CATG1(3),CATG2(3),CATG3(3),CATG4(3),CATG5(3),CATG
000015          X            TWAIT  (ALL CASES)       WEEKENDS      -   *.**     *.**
000016           X     MEEN(4),STDEV(4),CATG1(4),CATG2(4),CATG3(4),CATG4(4),CATG5(4),CATG
000017          X            TVEC   (ALL CASES)       WEEKDAYS      -   *.**     *.**
000018           X     MEEN(5),STDEV(5),CATG1(5),CATG2(5),CATG3(5),CATG4(5),CATG5(5),CATG
000019          X            TVEC   (ALL CASES)       WEEKENDS      -   *.**     *.**
000020           X     MEEN(6),STDEV(6),CATG1(6),CATG2(6),CATG3(6),CATG4(6),CATG5(6),CATG
000021          X            TWAIT-TOL (ALL CASES)    WEEKDAYS      -   *.**     *.**
000022           X     MEEN(7),STDEV(7),CATG1(7),CATG2(7),CATG3(7),CATG4(7),CATG5(7),CATG
000023          X            TWAIT-TOL (ALL CASES)    WEEKENDS      -   *.**     *.**
000024           X     MEEN(8),STDEV(8),CATG1(8),CATG2(8),CATG3(8),CATG4(8),CATG5(8),CATG
000025          X            DEMERIT (ALL CASES)      WEEKDAYS      -   *.**     *.**
000026           X     MEEN(9),STDEV(9),CATG1(9),CATG2(9),CATG3(9),CATG4(9),CATG5(9),CATG
000027          X            DEMERIT (ALL CASES)      WEEKENDS      -   *.**     *.**
000028           X     MEEN(10),STDEV(10),CATG1(10),CATG2(10),CATG3(10),CATG4(10),CATG5(1
000029          X            TSVC   (ALL CASES)       WEEKDAYS      -   *.**     *.**
000030           X     MEEN(11),STDEV(11),CATG1(11),CATG2(11),CATG3(11),CATG4(11),CATG5(1
000031          X            TSVC   (ALL CASES)       WEEKENDS      -   *.**     *.**
000032           X     MEEN(12),STDEV(12),CATG1(12),CATG2(12),CATG3(12),CATG4(12),CATG5(1
000033                END
000034
000035          IBUTIONS                                                          13
000036                                                                            11
000037          -------------------PERCENT------------------------------------- 1
000038          -0.5   0.5-1.0   1.0-2.0   2.0-3.0   3.0-4.0   4.0-5.0   5.0-10.0   >10   1
000039          **      ***       ***       ***       ***       ***       ***       **    1
000040          6(1),CATG7(1),CATG8(1)
000041          **      ***       ***       ***       ***       ***       ***       **    13
000042          6(2),CATG7(2),CATG8(2)
000043                                                                            11
000044          ------------------HOURS--------------------------------------- 1
000045          -0.5   0.5-1.0   1.0-2.0   2.0-3.0   3.0-4.0   4.0-5.0   5.0-10.0   >10   1
000046          **      ***       ***       ***       ***       ***       ***       **    1
000047          6(3),CATG7(3),CATG8(3)
000048          **      ***       ***       ***       ***       ***       ***       **    1
000049          6(4),CATG7(4),CATG8(4)
000050          **      ***       ***       ***       ***       ***       ***       **    1
000051          6(5),CATG7(5),CATG8(5)
000052          **      ***       ***       ***       ***       ***       ***       **    1
000053          6(6),CATG7(6),CATG8(6)
000054          **      ***       ***       ***       ***       ***       ***       **    1
```

```
000055          6(7),CATG7(7),CATG8(7)
000056          **      ***     ***     ***     ***     ***     ***     **   1
000057          6(8),CATG7(8),CATG8(8)
000058          **      ***     ***     ***     ***     ***     ***     **   1
000059          6(9),CATG7(9),CATG8(9)
000060          **      ***     ***     ***     ***     ***     ***     **   1
000061          0),CATG6(10),CATG7(10),CATG8(10)
000062          **      ***     ***     ***     ***     ***     ***     **   1
000063          1),CATG6(11),CATG7(11),CATG8(11)
000064          **      ***     ***     ***     ***     ***     ***     **   1
000065          2),CATG6(12),CATG7(12),CATG8(12)
000066                END
```

```
000001                SUBROUTINE JUMPER
000002                READ (5,1) N,A,B
000003          1     FORMAT(I2,2A6)
000004                WRITE(6,2)N,A,B
000005          2     FORMAT(1H1,T52,'SARSIM - DISTRICT',I2,32X,'DATE ',2A6)
000006                READ(5,3)A,B,N
000007          3     FORMAT(2A5,I2)
000008                WRITE(6,4)A,B,N
000009          4     FORMAT(1H0,T54,2A6,2X,'10',I2)
000010                READ (5,5)N,M
000011          5     FORMAT(2I2)
000012                WRITE(6,6)N,M
000013          6     FORMAT(1H0,T51,'FORECAST FROM 19',I2,2X,'TO',2X,I2)
000014                READ(5,8)N
000015          8     FORMAT(I2)
000016                IF(N) 300,300,301
000017          301   WRITE(6,7)
000018          7     FORMAT(T60,'WITH')
000019                DO 9 I=1,N
000020          11    FORMAT(T49,F5.1,' PCT CHANGE FOR ',3A6)
000021                READ(5,10)A,B,C,D
000022          10    FORMAT(F5.1,3A6)
000023                WRITE(6,11)A,B,C,D
000024          9     CONTINUE
000025          300   WRITE(6,12)
000026          12    FORMAT(1H0,T51,'I.  INPUT CONDITIONS')
000027                WRITE(6,13)
000028          13    FORMAT(1H ,T12,'A. STATION CHANGES')
000029                WRITE(6,14)
000030          14    FORMAT(T15,'1. ADD')
000031                READ(5,15)N
000032          15    FORMAT(I2)
000033                IF(N)35,35,36
000034          36    DO 16 III=1,N
000035                READ(5,18)K,A,B,I,J,L,M,IA,IB,IC,ID,IE,IF,IG,IJ,IK
000036          18    FORMAT(I3,2F8.1,13I3)
000037                WRITE(6,17)K,A,B,I,J,L,M,IA,IB,IC,ID,IE,IF,IG,IJ,IK
000038          17    FORMAT(T18,'OPFAC ',I3,2X,'XS = ',F8.1,'YS = ',F8.1,' ACS = ',I3,'
000039               1,',I3,',',I3,',',I3,2X,'ADJS = ',I3,',',I3,',',I3,',',I3,2X,'CUT =
000040               2 ',I3,',',I3,',',I3,',',I3,',',I3)
000041          16    CONTINUE
000042                GO TO 55
000043          35    WRITE(6,51)
000044          51    FORMAT(T18,'NO ADDITIONS')
000045          55    WRITE(6,19)
000046          19    FORMAT(T15,'2. DELETE')
000047                READ(5,20)N
000048          20    FORMAT(I2)
000049                IF(N)21,21,22
000050          22    DO 25 I=1,N
000051                READ(5,23)K
000052          23    FORMAT(I3)
000053                WRITE(6,24)K
000054          24    FORMAT(T18,'OPFAC ',I3)
```

```
000055          25    CONTINUE
000056                GO TO 57
000057          21    WRITE(6,56)
000058          56    FORMAT(T18,'NO DELETIONS')
000059          57    WRITE(6,26)
000060          26    FORMAT(T15,'3. ATTRIBUTE CHANGE')
000061                READ(5,27)N
000062          27    FORMAT(I2)
000063                IF(N)28,28,29
000064          29    DO 30 I=1,N
000065                READ(5,31)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000066          31    FORMAT(I3,12A6)
000067                WRITE(6,32)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000068          32    FORMAT(T18,'OPFAC ',I3,3X,12A6)
000069          30    CONTINUE
000070                GO TO 59
000071          28    WRITE(6,58)
000072          58    FORMAT(T18,'NO ATTRIBUTE CHANGES')
000073          59    WRITE(6,33)
000074          33    FORMAT(1H ,T12,'B.  RESOURCE TYPE CHANGES')
000075                WRITE(6,42)
000076          42    FORMAT(T15,'1. NEW  (SEE INITIALIZATION DATA FOR CHANGES TO CAPABI
000077               1LITY MATRIX(CPBL))')
000078                READ(5,34)N
000079          34    FORMAT(I2)
000080                IF(N)37,37,38
000081          38    DO 41 I=1,N
000082                READ(5,39)K,A1,A2,A3,A4,A5,A6,A7,A8
000083          39    FORMAT(I2,2F7.1,F5.2,F4.1,3F5.1,F6.1)
000084                WRITE(6,40)K,A1,A2,A3,A4,A5,A6,A7,A8
000085          40    FORMAT(T18,'RST=',I2,6X,'COSTD=',F7.1,3X,'END=',F7.1,2X,'MR=',F5.2
000086               1,2X,'SLIM=',F4.1,2X,'SOA1=',F5.1,2X,'SOA2=',F5.1,2X,'SOA3=',F5.1,2
000087               2X,'TF=',F6.1)
000088          41    CONTINUE
000089                GO TO 60
000090          37    WRITE(6,61)
000091          61    FORMAT(T18,'NO NEW ADDITIONS')
000092          60    WRITE(6,43)
000093          43    FORMAT(T15,'2. ATTRIBUTE CHANGE')
000094                READ(5,44)N
000095          44    FORMAT(I2)
000096                IF(N)45,45,46
000097          46    DO 49 I=1,N
000098                READ(5,47)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000099          47    FORMAT(I2,12A6)
000100                WRITE(6,48)K,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12
000101          48    FORMAT(T18,'RST',I2,6X,12A6)
000102          49    CONTINUE
000103                GO TO 63
000104          45    WRITE(6,50)
000105          50    FORMAT(T18,'NO ATTRIBUTE CHANGES')
000106          63    WRITE(6,64)
000107          64    FORMAT(T15,'3. CAPABILITY CHANGE')
000108                READ(5,66)N
000109          66    FORMAT(I2)
000110                IF(N)67,67,68
000111          68    WRITE(6,65)
000112          65    FORMAT(T18,'SEE INITIALIZATION DATA FOR CHANGES TO CAPABILITY MATR
```

```
000113          1IX(CPBL) FOR,')
000114           DO 69 I=1,N
000115           READ(5,70)K
000116      70   FORMAT(I3)
000117           WRITE(6,71)
000118      71   FORMAT(T18,'PST=',I3)
000119      69   CONTINUE
000120           GO TO 72
000121      67   WRITE(6,73)
000122      73   FORMAT(T18,'NO CAPABILITY CHANGES')
000123      72   WRITE(6,74)
000124      74   FORMAT(1H ,T12,'C.  RESOURCE INVENTORY CHANGES')
000125           WRITE(6,75)
000126      75   FORMAT(T15,'1. NEW OPFACS')
000127           READ(5,76)N
000128      76   FORMAT(I2)
000129           IF(N)77,77,78
000130      78   DO 79 I=1,N
000131           READ(5,80)A,B,C,D,E,F,G,H,O,P,Q,R
000132      80   FORMAT(12A6)
000133           WRITE(6,81)A,B,C,D,E,F,G,H,O,P,Q,R
000134      81   FORMAT(T18,12A6)
000135      79   CONTINUE
000136           GO TO 82
000137      77   WRITE(6,83)
000138      83   FORMAT(T18,'NO NEW OPFACS')
000139      82   WRITE(6,84)
000140      84   FORMAT(T15,'2. EXISTING OPFACS')
000141           READ(5,85)N
000142      85   FORMAT(I2)
000143           IF(N)86,86,87
000144      87   DO 88 I=1,N
000145           READ(5,89)A,B,C,D,E,F,G,H,O,P,Q,R
000146      89   FORMAT(12A6)
000147           WRITE(6,90)A,B,C,D,E,F,G,H,O,P,Q,R
000148      90   FORMAT(T18,12A6)
000149      88   CONTINUE
000150           GO TO 91
000151      86   WRITE(6,92)
000152      92   FORMAT(T18,'NO RESOURCE CHANGES TO  EXISTING OPFACS')
000153      91   WRITE(6,93)
000154      93   FORMAT(1H ,T12,'D.  CREW MANNING LEVEL CHANGES')
000155           WRITE(6,94)
000156      94   FORMAT(T15,'1. NEW OPFACS')
000157           READ(5,95)N
000158      95   FORMAT(I2)
000159           IF(N)96,96,97
000160      97   DO 125 I=1,N
000161           READ(5,98)N1,N2,N3,N4,N5,N6,N7,N8,N9
000162      98   FORMAT(9I3)
000163           WRITE(6,99)N1,N2,N3,N4,N5,N6,N7,N8,N9
000164      99   FORMAT(T18,'OPFAC ',I3,2X,'SHFT1=',I3,2X,'SHFT2=',I3,2X,'SHFT3=',I
000165          13,2X,'SHFT4=',I3,2X,'SHFT5=',I3,2X,'SHFT6=',I3,2X,'SHFT7=',I3,2X,'
000166          2SHFT8=',I3)
000167     125   CONTINUE
000168           GO TO 100
000169      96   WRITE(6,101)
000170     101   FORMAT(T18,'NO CHANGES')
```

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS